VMS/ULTRIX Connection

digital

System Manager's Guide

System Manager's Guide

Order Number: AA-LU50C-TE

VMS/ULTRIX Connection System Manager's Guide

AA–LU50C–TE

September 1990

This guide describes the procedures for setting up and managing the VMS/ULTRIX Connection.

Revision/Update Information: This revised manual supersedes the *VMS/ULTRIX Connection System Manager's Guide*, Order No. AA–LU50A–TE.

Operating System and Version: VMS Version 5.3 or higher

Software Version: VMS/ULTRIX Connection Version 1.3 or higher

digital equipment corporation
maynard, massachusetts

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause of DFARS 252.227-7013.

© Digital Equipment Corporation 1990
All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital
CDA
DDIF
DDIS
DEC
DECnet
DECstation
DECsystem
DECUS
DECwindows
DTIF
MASSBUS
MicroVAX
Q-bus
ULTRIX
ULTRIX Mail Connection
ULTRIX Worksystem Software
UNIBUS
VAX
VAXstation
VMS
VMS/ULTRIX Connection
VT
XUI

Ethernet is a registered trademark of Xerox Corporation.

Network File System and NFS are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark of AT&T in the USA and other countries.

Contents

Preface xi

Summary of Technical Changes xv

1 Introduction

1.1 Functional Components 1–1
1.1.1 Internet Network 1–2
1.1.2 Network File System 1–3
1.1.3 File Transfer Protocol 1–3
1.1.4 Telnet 1–3
1.1.5 rlogin Server 1–3
1.1.6 BIND Resolver 1–3
1.1.7 Programming Interface 1–3
1.2 Using the Connection Software Commands 1–4

2 Internet Concepts

2.1 Overview 2–1
2.2 Client-Server Model 2–3
2.3 Ports 2–3
2.4 Internet Protocols 2–4
2.4.1 Internet Protocol 2–5
2.4.2 Transmission Control Protocol 2–5
2.4.3 User Datagram Protocol 2–5
2.4.4 Internet Control Message Protocol 2–5
2.4.5 Address Resolution Protocol 2–6

2.4.6 Routing Information Protocol 2–6
2.4.6.1 How Dynamic Routing Works 2–6
2.4.7 File Transfer Protocol 2–8
2.4.8 Telnet Protocol 2–8
2.5 Berkeley Internet Name Domain Resolver 2–8
2.6 Internet Addresses 2–9
2.6.1 Address Notation 2–10
2.6.2 Network Classes 2–10
2.6.3 Network Mask 2–12
2.6.4 Broadcast Mask 2–14
2.7 Routing 2–16
2.8 Fragmentation 2–19

3 Internet Management

3.1 Modifying VMS SYSGEN Parameters 3–2
3.1.1 Increasing the Nonpaged Pool Size 3–2
3.1.2 Increasing Global Pages and Global Sections 3–3
3.2 Configuring the Internet 3–3
3.2.1 Executing the Procedure 3–4
3.2.2 Configuring the Internet Parameters 3–5
3.2.2.1 Defining Multi-Homed Hosts 3–9
3.2.2.2 Defining Hosts in VAXclusters 3–11
3.3 Starting and Stopping the Internet Software 3–12
3.3.1 Starting the Internet 3–12
3.3.2 Shutting Down the Internet 3–12
3.4 Setting Up the Internet Databases 3–13
3.4.1 Host Database 3–13
3.4.2 Network Database 3–14
3.4.3 Route Database 3–15
3.5 Using a VMS Host as Gateway Between Networks 3–16
3.6 Tuning the System for Use with Internet 3–17
3.6.1 Nonpaged Pool Requirements 3–18
3.6.1.1 Memory Buffer Space Requirements 3–18
3.6.1.2 Device-Socket Space Requirements 3–19
3.6.2 Broadcast Messages 3–20
3.6.3 Checksum Calculations 3–20
3.6.4 IP Reassembly Time 3–21
3.6.5 Socket Buffer Size for UDP and TCP 3–21
3.6.6 I/O Request Packets 3–22
3.7 Creating the Internet Interfaces 3–23
3.7.1 Internet Pseudo-Interface 3–23
3.7.1.1 Extending Subnet Routing 3–24
3.8 Using a VAXcluster System as a Server 3–25

3.9 Using the Address Resolution Protocol 3–26
3.10 Displaying Control and Statistical Information 3–26
3.10.1 SHOW COMMUNICATION Command 3–27
3.10.2 SHOW PROTOCOL Command 3–29
3.10.3 SHOW INTERFACE Command 3–31
3.10.4 SHOW DEVICE_SOCKET Command 3–32
3.10.5 SHOW HOST Command 3–33
3.10.6 SHOW ARP Command 3–34
3.11 Managing Dynamic Routing 3–35
3.12 Managing the BIND Resolver 3–36
3.13 Managing FTP 3–36
3.13.1 Starting the FTP Server 3–36
3.13.2 Shutting Down the FTP Server 3–37
3.13.3 Defining Logical Names for FTP 3–37
3.14 Managing the Telnet and rlogin Server 3–38
3.14.1 Tuning Telnet Server Parameters 3–38
3.14.2 Tuning Internet Parameters for Telnet 3–40
3.14.3 Monitoring Telnet 3–41
3.14.4 Managing Requests That Cannot Be Satisfied 3–42
3.15 Troubleshooting the Connection Internet Software 3–42
3.15.1 Troubleshooting BIND 3–42

4 Network File System Server

4.1 Overview 4–1
4.1.1 VMS and ULTRIX file System 4–2
4.1.2 Connection Management Utility 4–2
4.1.3 NFS Server 4–3
4.1.4 Connection File System 4–4
4.1.4.1 Connection VMS File System 4–4
4.1.4.2 Connection ULTRIX File System 4–5
4.1.4.3 Selecting the Correct File System for an NFS Client 4–5
4.1.5 Connection File System Run-Time Library 4–6
4.2 Managing the NFS Server 4–6
4.2.1 Export Database 4–7
4.2.2 Proxy Database 4–7
4.2.2.1 Superuser (Root) Account 4–9
4.2.2.2 Client Registration 4–9
4.3 Setting Up and Creating File Systems 4–10
4.4 Setting Up a Connection ULTRIX File System 4–10
4.5 Creating a Connection ULTRIX File System 4–10
4.5.1 Creating an ULTRIX File System 4–11
4.5.2 Binding the VMS File System 4–11
4.5.3 Binding the ULTRIX File System 4–11

4.5.4 Creating a Top-Level Directory for Each ULTRIX NFS Client 4–12
4.5.5 Exporting a Mountable ULTRIX File System or Directory 4–13
4.6 Security Considerations 4–13
4.7 Maintaining the Connection ULTRIX File System 4–15
4.7.1 Removing Links to a File 4–16
4.7.2 Removing Links to a Directory 4–16
4.7.3 Deleting a Connection ULTRIX File System 4–16
4.7.4 Copying Files into Connection ULTRIX File Systems 4–16
4.7.5 Verifying the Integrity of a Connection ULTRIX File System 4–17
4.7.6 Backingup NFS-mounted Files 4–18
4.7.7 Restoring NFS-mounted Files 4–19
4.7.7.1 Restoring ULTRIX File Systems 4–19
4.8 Starting the NFS Server 4–20
4.9 Tuning NFS Server Performance 4–21
4.9.1 UAF File Limits 4–30
4.9.2 Displaying NFS Server Tuning Information 4–31
4.9.2.1 Monitoring System Services 4–33
4.9.2.2 Monitoring Connection File System Performance 4–34
4.10 Stopping the NFS Server 4–37
4.11 Troubleshooting the NFS Server 4–38
4.12 Running the NFS Server on a VMS Internet Cluster 4–41

5 VMS/ULTRIX Connection Commands

5.1 Connection File System Commands 5–1
5.2 Internet Commands 5–2
5.3 NFS Commands 5–2
5.4 Database Commands 5–3
5.5 Miscellaneous Commands 5–4
5.6 UCX Command Reference 5–4
ADD EXPORT 5–5
ADD PROXY 5–7
ANALYZE CONTAINER 5–9
BIND 5–11
CONVERT/VMS HOST 5–13
CONVERT/VMS NETWORK 5–15
CONVERT/VMS PROXY 5–17
CONVERT/ULTRIX HOST 5–19
CONVERT/ULTRIX NETWORK 5–21
CREATE CONTAINER 5–23
CREATE DIRECTORY 5–27

CREATE EXPORT 5–31
CREATE HOST 5–32
CREATE NETWORK 5–33
CREATE PROXY 5–34
CREATE ROUTE 5–35
DELETE CONTAINER 5–36
DIRECTORY 5–37
EXPORT 5–40
IMPORT 5–41
LOOP 5–45
REMOVE DIRECTORY 5–47
REMOVE EXPORT 5–48
REMOVE FILE 5–50
REMOVE PROXY 5–51
SET ARP 5–53
SET COMMUNICATION 5–56
SET HOST 5–61
SET INTERFACE 5–63
SET NAME_SERVICE 5–69
SET NETWORK 5–72
SET NFS_SERVER 5–74
SET ROUTE 5–77
SHOW ARP 5–81
SHOW BIND 5–83
SHOW CFS 5–84
SHOW COMMUNICATION 5–87
SHOW DEVICE_SOCKET 5–89
SHOW EXPORT 5–92
SHOW HOST 5–94
SHOW INTERFACE 5–98
SHOW NAME_SERVICE 5–100
SHOW NETWORK 5–101
SHOW NFS_SERVER 5–103
SHOW PROTOCOL 5–105
SHOW PROXY 5–107
SHOW ROUTE 5–110
START COMMUNICATION 5–113
START SERVICE 5–114
START ROUTING 5–116

STOP COMMUNICATION 5–118
STOP ROUTING 5–120
STOP SERVICE 5–121
UNBIND 5–123
ZERO INTERFACE 5–124
ZERO NFS_SERVER 5–126
ZERO PROTOCOL 5–128

A Connection Error Messages

A.1 Display Terminal Errors Messages A–1
A.2 File System Messages Directed to the Operator's Console A–31
A.3 Internet Messages Directed to the Operator's Console A–38

B ANALYZE CONTAINER Validation Messages

B.1 ANALYZE CONTAINER Summary Information B–2
B.2 General Information Messages B–2
B.3 Superblock Validation Messages B–4
B.3.1 Inode Validation Messages B–8
B.4 Directory and Raw Data Validation Messages B–14
B.5 Bitmap and Deleted Cells Validation Messages B–19
B.6 Directory Hierarchy Reconstruction Messages B–22

Glossary

Index

Examples

3–1 SHOW COMMUNICATION 3–28
3–2 SHOW PROTOCOL Display 3–30
3–3 SHOW INTERFACE 3–32
3–4 SHOW DEVICE_SOCKET Display 3–33
3–5 SHOW HOST Display 3–34
3–6 SHOW ARP Display 3–35
4–1 UCX$NFS_STARTUP.COM File 4–21
4–2 SHOW NFS_SERVER Command Display 4–32

4–3 SHOW CFS Command Display 4–34
4–4 SHOW CFS/SUMMARY Command Display 4–35
4–5 UCX$NFS_SHUTDOWN.COM File 4–37

Figures

1–1 Connection Software Functional Components 1–2
2–1 Internet Network Configuration 2–2
2–2 Port Number Ranges 2–4
2–3 Internet Network Classes 2–11
2–4 Network Masks 2–14
2–5 Internet Routing 2–17
2–6 Subnet Routing on a Single Physical Network 2–18
2–7 Subnet Routing on Multiple Physical Networks 2–19
3–1 Sample Network with a Gateway 3–17
4–1 A Client's File System 4–2
4–2 Connection NFS Software Components 4–3
4–3 Connection File System 4–5
4–4 Exported File System 4–15
4–5 Exporting and Importing Files 4–17

Tables

2–1 Top-Level Domains 2–9
2–2 Network Number Ranges 2–12
2–3 Broadcast Addresses 2–15
3–1 Internet Databases 3–13
3–2 Queue Size Quotas for Device-Sockets 3–21
3–3 Interface Names 3–23
3–4 Displaying Control and Statistical Information 3–26
3–5 FTP Logical Names 3–37
4–1 Connection ULTRIX File System Components Analyzed 4–18
4–2 Logging Error Messages 4–25
4–3 Security Logical Name Mask Bits 4–27
4–4 File-Attributes Cache Values 4–36
4–5 Services and Resources Allocation Values 4–36
4–6 Mapping VMS System Error Messages to NFS Error Messages . . 4–40
5–1 Connection Databases 5–3

Preface

This guide describes the procedures for setting up and managing the VMS/ULTRIX Connection.

Intended Audience

This guide is intended for the experienced VMS system manager who must manage a VMS server with the VMS/ULTRIX Connection software installed. It describes how to perform network functions to control, monitor, and test Internet software running on a VMS operating system, as well as controlling VMS resources, maintaining server performance, and troubleshooting server problems.

Structure of This Document

This guide contains five chapters, two appendixes, and a glossary.

- Chapter 1 provides an overview of the VMS/ULTRIX Connection product.
- Chapter 2 describes Internet concepts: the client-server model, ports, protocols, addressing (including subnet addressing), routing, fragmentation, and the Berkeley Internet Name Domain resolver.
- Chapter 3 describes how to manage the Connection Internet and how to set the Internet parameters for the best performance.
- Chapter 4 describes how to manage the Connection Network File System (NFS) server. It describes how to set up Connection VMS and ULTRIX file systems for the NFS clients and how to tune the NFS server for best performance.
- Chapter 5 provides reference information for all the Connection commands (Internet, Connection File System, and Connection NFS management).

- Appendix A lists the error codes for the Connection.
- Appendix B lists the messages that can be generated while you are creating a container file.

Associated Documents

The VMS/ULTRIX Connection documentation set also includes the following books:

- The *VMS/ULTRIX Connection Installation Guide* describes how to install the Connection software.
- The *VMS/ULTRIX Connection Programming Manual* describes how to write Internet network applications by using the QIO interface and VAX C socket interface.
- The *VMS/ULTRIX Connection User's Guide* provides user information on FTP, Telnet, and NFS.

The following Digital remote procedure call (DECrpc) documents are included with the VMS/ULTRIX Connection documentation:

- The *Guide to the Location Broker*
- The *DECrpc Programming Guide*

You can also order *Internetworking with TCP/IP: Principles, Protocols, and Architecture* by Douglas Comer through Digital (Order number: ER–TCPIP–TM–001). This book provides an introduction and overview of Internet concepts as well as an explanation of the various protocols, Internet addressing, and other Internet concepts you may need to understand to manage the VMS/ULTRIX Connection software.

Conventions Used in This Document

The following conventions are used throughout this manual:

Convention	Meaning
Ctrl/C	A key combination, shown with a slash separating two key names, indicates that you hold down the first key while you press the second key. For example, the key combination Ctrl/C indicates that you hold down the key labeled Ctrl while you press the key labeled C. In examples, a key combination is enclosed in a box.
$ SHOW TIME 05-JUN-1989 11:55:22	In examples, system output (what the system displays) is shown in black. User input (what you enter) is shown in red.
$ TYPE MYFILE.DAT . . .	In examples, a vertical series of periods, or ellipsis, means either that not all the data that the system would display in response to a command is shown or that not all the data a user would enter is shown.
input-file, . . .	In command syntax or examples, a horizontal ellipsis indicates that additional parameters, values, or other information can be entered, that preceding items can be repeated one or more times, or that optional arguments in a statement have been omitted.
[logical-name]	Brackets indicate that the enclosed item is optional. (Brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.
A **host** sends	Terms that appear in text in bold print are defined in the glossary.
/usr/smith/work	Italics is used to show ULTRIX commands and syntax.

Summary of Technical Changes

The following list briefly describes the changes to the VMS/ULTRIX Connection, Version 1.3. Many of these features are documented in the VMS/ULTRIX Connection documentation. For more information see the Release Notes for Version 1.3.

- Major functional enhancements

  Version 1.3 includes the following major enhancements:

  - Support for the Berkeley Internet Name Domain (BIND) resolver
  - Support for dynamic routing
  - Support for Digital remote procedure call (DECrpc), Version 1.0

- Changes to the Connection Management include the following:

  - The SET NAME_SERVICE command has been added. This command modifies the system and process image parameters for the BIND resolver.
  - The SHOW NAME_SERVICE command has been added. This command displays all information pertaining to the name service.
  - The START ROUTING command has been added. This command starts dynamic routing.
  - The STOP ROUTING command has been added. This command stops dynamic routing.
  - The SPAWN command has been removed.
  - The SET COMMUNICATION command has been modified to include the /[NO]BROADCAST qualifier.
  - The SET ROUTE command has been modified to include the /DEFAULT and /PERMANENT qualifiers.

- The SHOW COMMUNICATION command has been modified to include the /ROUTE qualifier.
- The SHOW EXPORT, SHOW HOST, SHOW NETWORK, SHOW PROXY, and SHOW ROUTE commands have been modified to include the /OUTPUT qualifier.
- The SHOW HOST command has also been modified to include the /DOMAIN, /[NO]LOCAL, /SERVER qualifiers.
- The SHOW ROUTE command has also been modified to include the /PERMANENT qualifier.
- The ADD PROXY and SHOW PROXY commands have been modified to include the /PERMANENT qualifier.

Other changes to Connection management include the following:

- To support dynamic routing, the UCX$ROUTE database is now located in SYS$COMMON:[SYSEXE] rather than SYS$SPECIFIC:[SYSEXE].
- The file format for UCX$ROUTE.DAT has changed. If you installed a previous version of the Connection, you must run UCX$CONFIG to convert your database to the new format.

- Changes to the Network File System (NFS) include the following:
  - The UCX$CFS_SHOW_VERSION logical has been added to the UCX$NFS_STARTUP.COM. This logical enables you to specify whether version numbers are displayed with the file names, when there is only one version of the file.
  - Support for Automount has been added. Automount enables you to implicitly mount file systems without specifying the mount command. This feature is transparent to the NFS server. It is documented in the ULTRIX Version 4.0 documentation.
  - Support for VMS network access control has been added.
  - The ondisk proxy database and the volatile are now synchronized.
  - NFS parameters are now logged to the error log file.
- Changes to the Internet include the following:
  - Support for dynamic routing has been added through the Connection's implementation of the routing information protocol (RIP).
  - Support for extending subnet routing has been added.
  - You can now use the Internet Cluster Alias while your host acts as a gateway.

- The /[NO]BROADCAST qualifier has been added to the UCX SET COMMUNICATION command. This qualifier allows the system manager to enable or disable the checking of the privileges required to send Internet broadcast packets.

- The Telnet Client has been changed as follows:

  The SET DEVICE/TERMINAL command has been added. This command enables you to specify the terminal type to the remote host.

- Changes to the Telnet Server include the following:
  - Login and logout messages can now be displayed on the operator's console.
  - Security audit has been added.

- Changes to the File Transfer Protocol (FTP) Client include the following:
  - The GET command has been modified to allow the use of wildcards in the remote file name.
  - The ENABLE PARSE and DISABLE PARSE commands have been added.
  - The /CONFIRM qualifier has been added to the GET and PUT commands. This qualifier enables you to confirm GET and PUT copy operations.

- The FTP Server has been changed as follows:

  The FTP server now executes LOGINOUT.EXE during child process creation.

- Changes to Interprocess Communications include the following:

  The following new INET_ACP call codes were added to the IO$_ACPCONTROL I/O function:
  - INETACPC$_C_HOSTENT — INET_ACP returns full host information in a HOSTENT structure.
  - INETACPC$_C_NETENT — INET_ACP returns full network information in a NETENT structure.

The following list briefly describes the changes to the VMS/ULTRIX Connection, Version 1.2. For more information see the Release Notes for Version 1.2.

- Support for the following has been added:
  - Telnet client and server
  - rlogin server
  - BSD socket programming interface

- Changes to the VMS/ULTRIX Connection system management include the following:
  - The /[NO]LOG and /UID qualifiers have been added to the CREATE CONTAINER, CREATE DIRECTORY, and IMPORT commands.
  - The IMPORT command has been changed to allow users to import files into container files without requiring SYSPRV or BYPASS privilege.
  - The /MODE qualifier has been added to the IMPORT command.
  - The /NOCLUSTER qualifier has been added to the SET INTERFACE command.
  - Command line recall has been added to UCX$UCP.
  - A SPAWN command has been added to UCX$UCP.
  - Wildcard support has been added to the SHOW BIND command.
  - The following qualifiers have been added to the SHOW DEVICE_ SOCKET command:

    /HOST
    /PORT
    /SERVICE
    /TYPE
  - The UCX SHOW DIRECTORY command has been enhanced to include the file specification.
  - The UCX SHOW DEVICE command has been enhanced to include Telnet and rlogin server information.
  - Counters have been added to show the current device-sockets and the peak number of device-sockets on your system.
- Changes to Network File System (NFS) include the following:
  - The NFS server now allows a client host to use a legitimate alias name to mount a file system and access files on the NFS server.
  - You can specify an Internet address for the client host name.
  - You can specify null names.
  - "Stale" file handle processing has been implemented.
- Changes to the Internet include the following:
  - You can issue nonblocking I/O as a modifier to the read or write QIOs. Additionally, two I/O subfunction masks, IO$M_NOW and IO$M_NOWAIT, have been added.

- The minimum value for UDP and TCP read/write byte quota for device-sockets has been set to the size of an Internet internal data buffer.

- Changes to File Transfer Protocol (FTP) Client include the following:
  - Certain FTP error messages have been changed to supply additional information.
  - The FTP client has been changed to default the user name to lowercase.
  - The FTP client's directory display has been improved.
  - Support for putting ASCII files to a line printer has been added.
  - Support for command line recall has been added to the FTP client's interface.
  - During file transfer, files are now opened in nonsharing mode.
  - The QUOTE command has been added to FTP client's user interface.
  - You can now enter a host name on the FTP command line.
  - Directory display performance has been enhanced.
- Changes to FTP Server include the following:
  - The FTP server now aborts control connections that have been idle for 15 minutes.
  - The FTP server now disallows logging in to a captive account.

1

Introduction

The VMS/ULTRIX Connection enables VMS hosts and UNIX hosts to communicate and share data. The Connection establishes a client-server relationship between UNIX hosts and VMS hosts. The Connection provides both client and server capabilities for most of its software components. The client-server relationship can be established with a single host or multiple hosts in a VAXcluster system.

Through the Connection, VMS hosts can provide multiple UNIX users with access to VMS files and ULTRIX file systems that reside on the VMS hosts. (A **host** is referred to as a node in VMS terminology.)

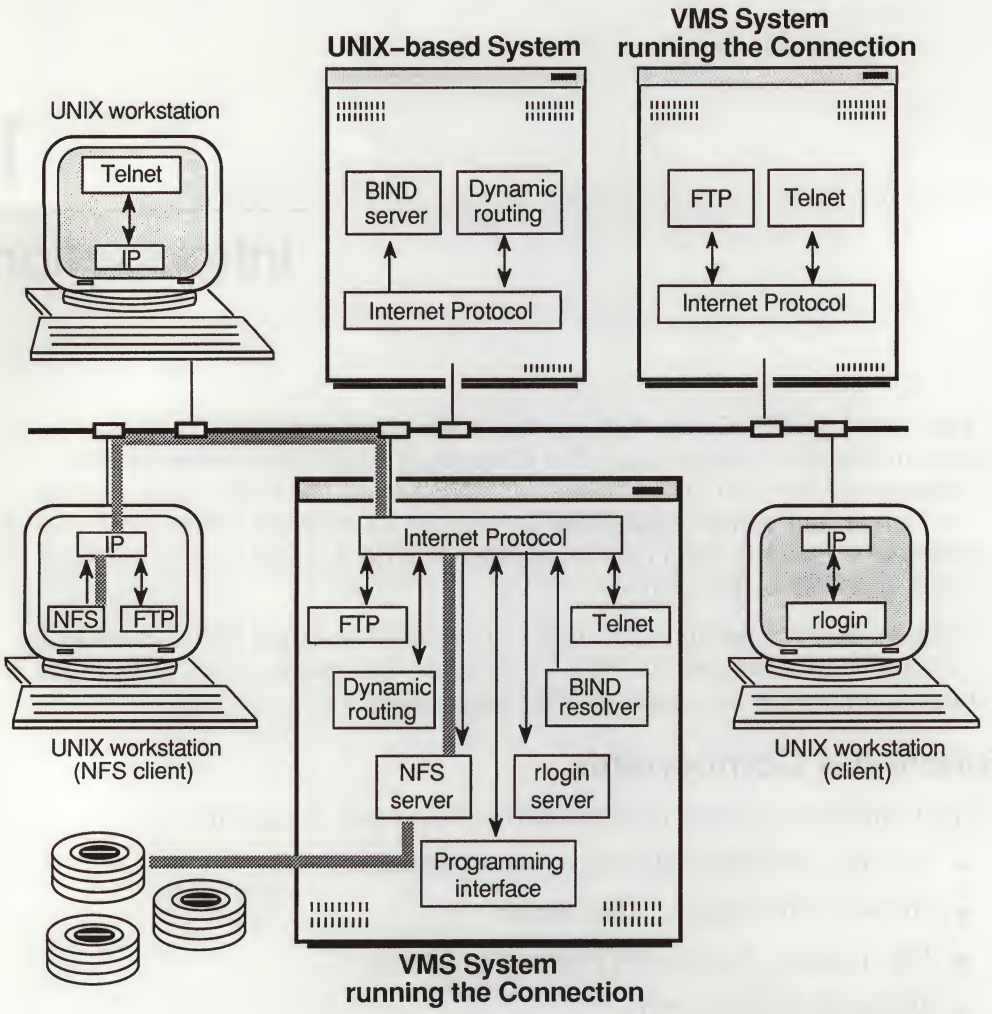
1.1 Functional Components

The Connection consists of the following functional components:

- Internet network (including dynamic routing)
- Network File System (NFS) server
- File Transfer Protocol (FTP) client and server
- Telnet client and server
- rlogin server
- Berkeley Internet Name Domain (BIND) resolver
- Programming interface

Figure 1–1 shows the relationships between these components.

Figure 1–1 Connection Software Functional Components



1.1.1 Internet Network

Clients communicate with the VMS system using the Internet networking protocols over an Ethernet medium. Communications between the clients and servers are not limited to the local area network; if the local network is connected by a host that serves as a gateway, communications can take place between the local network and other networks.

1.1.2 Network File System

The **Network File System NFS** provides the UNIX workstation users with transparent remote data sharing. This means that UNIX workstation users can directly access data that resides on the VMS server.

The UNIX user accesses not only VMS files but also ULTRIX file systems that reside on the VMS server. The ULTRIX file system is fully ULTRIX compatible, including full semantic compatibility.

1.1.3 File Transfer Protocol

The **File Transfer Protocol FTP** allows files to be transferred from one host to another host within a heterogeneous environment. It allows physical copies of data to be transferred from a VMS server to an FTP client or from an FTP client to the VMS server.

1.1.4 Telnet

The Telnet utility enables you to access any system on your network that supports the Telnet protocol. When you access Telnet, it establishes a virtual terminal connection between your terminal and the specified host. Once a connection is established with a remote host, it appears as if your terminal is connected directly to that host.

1.1.5 rlogin Server

The rlogin server enables UNIX clients to communicate with VMS hosts as if their terminals were connected directly to the host. Once a connection is established with the remote host, it appears as if the user is directly connected to that system.

1.1.6 BIND Resolver

The **Berkeley Internet Name Domain (BIND)** service is a host name and address lookup service for the Internet network. The BIND service is implemented in a client-server model. The client software is referred to as the resolver. The resolver allows client systems to obtain host names and addresses from servers rather than from locally hosted databases. As a result, you can use the BIND service to supplement the host address mapping provided by the local UCX$HOST file.

1.1.7 Programming Interface

The programming interface allows network application programmers to write programs that communicate over the Internet network. The *VMS/ULTRIX Connection Programming Manual* provides information on how to write network application programs.

1.2 Using the Connection Software Commands

The Connection provides commands to manage and tune the VMS server that is running the Connection software. These commands are described in Chapter 5.

The Connection management commands are implemented by a single VMS image, SYS$SYSTEM:UCX$UCP.EXE. The top-level DCL command used to invoke this image is UCX:

```
$
$ UCX
UCX> command
```

The UCX control program is linked against several shareable images. These images must be installed before any UCX commands can be executed. The images are installed by the command procedure SYS$MANAGER:UCX$UCP_STARTUP.COM. This command procedure is automatically invoked when the Connection software is started with SYS$MANAGER:UCX$STARTUP.

Note *DCL commands and parameters are not normally case sensitive. You must be aware of case sensitivity in supporting UNIX clients. For example, there is a difference between the uppercase host name "JUNE" and the lowercase name "june".*

Use quotes to preserve case sensitivity in your Connection commands. For example, to display the Internet address for host june, *issue the following command:*

```
UCX> SHOW HOST "june"
```

2

Internet Concepts

This chapter describes the following Internet concepts that are helpful in managing the VMS/ULTRIX Connection VMS server:

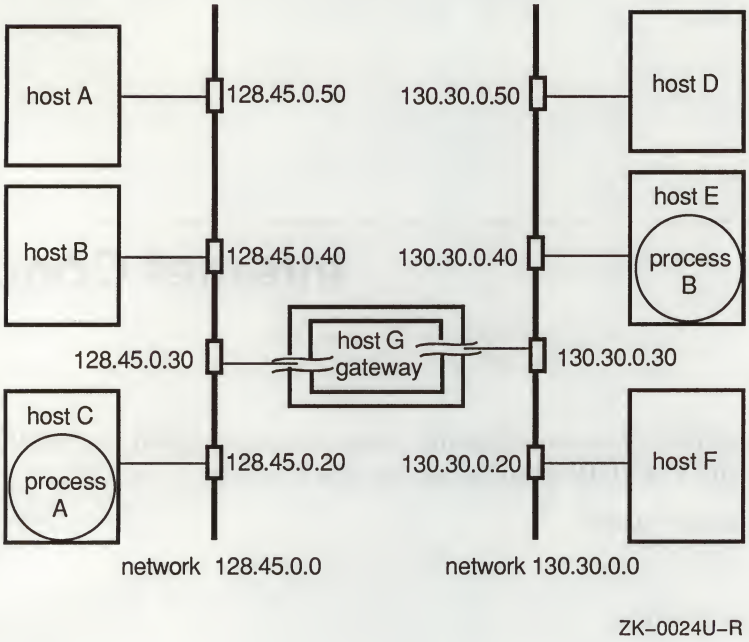
- Client-server model
- Ports
- Internet protocols
- Internet addresses
- Routing
- Fragmentation
- Berkeley Internet Name Domain resolver

2.1 Overview

An **Internet network** consists of two or more local area networks that are connected by a computer system that acts as a gateway. A **gateway** allows data to be transferred from one computer system to another computer system that is located on a different local area network. Figure 2–1 shows a possible Internet network configuration.

A **local area network** consists of two or more computer systems connected by an Ethernet communication medium. Each host computer connects to the transmission medium by a hardware interface connected to only one local area network.

Figure 2–1 Internet Network Configuration



Each host in an Internet network is identified by a unique host name and an Internet network address. An Internet address is made up of a network number field and a host number field. Each local area network has a unique network number, and each host on that network has a unique host number.

Data can be sent from one host to another by numerous routes. The VMS/ULTRIX Connection supports dynamic routing. If you enable dynamic routing, your host can update its routing database with routing information supplied by gateways over the network. For a Connection gateway to supply its routing database information to other hosts and networks, you must enable them with the UCX command START ROUTING/SUPPLY. If dynamic routing is not enabled, you must specify the routes to hosts that are not on the source local area network.

A network can be logically divided into several subnetworks. Subnetworks are useful for organizing hosts within a network into logical groups. These logical groups expand the network name space of an Internet address and allow several logical networks to exist in the same physical network or several physical networks. This is accomplished through the use of a network mask.

A Connection VMS server can be a single host, a whole VAXcluster system, or some hosts in a VAXcluster system. A whole VAXcluster system or some of the hosts in a VAXcluster system can be represented by a special alias called the VAXcluster alias. A **VAXcluster alias** appears to other hosts in the network to identify an actual host. Thus a remote host can address the cluster of hosts as a single host as well as any cluster member individually.

2.2 Client-Server Model

Host-to-host communication takes place between two processes. A process is a program that has been scheduled by system software to execute on a host; the host provides the context in which an image executes. Any process that offers a service over the network to another process is known as a **server**. Servers accept requests from other processes known as clients. A **client** sends a request and waits for the result from the server.

A VMS process uses an **Internet pseudodevice** to interface with the Internet protocols. The Internet pseudodevice contains standard VMS device information. The Internet pseudodevice driver maintains communication-specific information in a structure known as a **socket**.

2.3 Ports

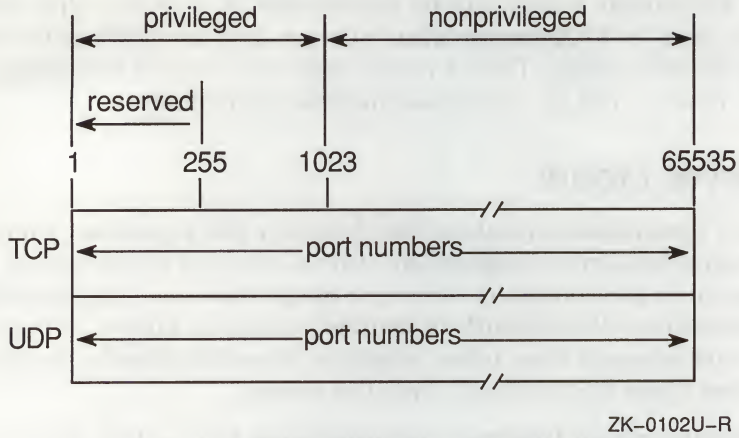
The endpoint of communication for a message is not a process name. Instead, each host contains a set of abstract endpoints called **ports**. Messages arriving for a particular port are queued until a process extracts them, and processes waiting at a port are blocked until messages arrive.

To communicate with a port on another host, a sender uses both the Internet address and the remote port number of the destination host. The Internet address identifies a particular network and host; the port number identifies the process on that host. With each message, the sender supplies a port number on the source machine to which replies should be addressed, making it possible for a recipient to reply to messages.

The local and remote ports do not usually use the same port number. The TCP/IP and UDP/IP protocols have the same range of port numbers. Figure 2–2 illustrates the port number ranges.

Port numbers in the range 1 to 1023 are considered to be **privileged ports**. Privilege means something different for each operating system. However, when a host receives a message from a privileged port, it can be assumed that the remote host has done some level of checking against the application using this port.

Figure 2–2 Port Number Ranges



The port numbers from 1 to 255 are **reserved** (assigned port numbers). The purpose of these assigned ports is to provide services to unknown callers by providing a service contact port. For example, FTP is assigned port numbers 20 (data) and 21 (control). Digital honors these assigned ports as implemented in both the Department of Defense (DoD) and the Defense Advanced Research Projects (DARPA) Internet communities.

The VMS operating system requires a process to have a privileged UIC, SYSPRV, or BYPASS privilege to bind to the local privileged ports (1 to 1023).

2.4 Internet Protocols

The VMS/ULTRIX Connection software supports the following Internet protocols:

- Internet Protocol (IP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)
- Routing Information Protocol (RIP)
- File Transfer Protocol (FTP)
- Telnet Protocol
- Berkeley Internet Name Domain (BIND) resolver

2.4.1 Internet Protocol

The **Internet Protocol (IP)** implements the mechanisms for connecting various networks and gateways into a system that can deliver packets from source to destination. IP is the protocol that insulates applications from needing to know network specifics. This protocol performs two major functions: internetwork addressing and fragmentation of messages.

The VMS/ULTRIX Connection provides support for IP trailer protocols on the receive operation. A trailer protocol is a protocol in which the protocol header follows the data written prior to it. Using trailer protocols has no impact on VMS/ULTRIX Connection performance.

2.4.2 Transmission Control Protocol

The **Transmission Control Protocol (TCP)** is a connection-oriented, end-to-end reliable protocol that functions as part of a layered hierarchy of protocols that support multinetwork applications. It provides for reliable interprocess communication between pairs of processes in host computers attached to distinct but interconnected computer communication networks. TCP does not require reliability of the communication protocols below itself. Therefore, TCP functions with lower-level protocols that are simple, potentially unreliable datagram services. TCP uses the Internet Protocol (IP) for a lower-level protocol with the Connection software.

The File Transfer Protocol (FTP) is an example of an application that uses TCP/IP.

2.4.3 User Datagram Protocol

The **User Datagram Protocol (UDP)** provides a datagram mode of communications within the environment of a computer network. UDP is used by applications that do not need a reliable stream service. Because UDP does not provide reliable service, some applications add error and sequence control to provide virtual circuits for reliability.

UDP requires that the Internet Protocol (IP) be used as the underlying protocol.

The Network File System (NFS) server is an example of an application that uses UDP/IP.

2.4.4 Internet Control Message Protocol

The **Internet Control Message Protocol (ICMP)** is a special-purpose protocol that gateways use to communicate with the network software in hosts. ICMP is a required part of the Internet Protocol (IP).

ICMP performs the following functions:

- Provides routing information

- Notifies hosts when a datagram cannot reach its destination
- Notifies hosts of datagram destruction caused by its time-to-live value reaching zero

2.4.5 Address Resolution Protocol

The **Address Resolution Protocol (ARP)** provides a dynamic mapping between Internet addresses and Ethernet physical addresses. This mapping is based on the Ethernet's ability to broadcast addresses. The originating host sends a broadcast packet that supplies the Internet address of the destined host and requests the Ethernet address of the destined host. The destined host receives the request and sends a reply that contains its physical Ethernet address. When the originating host receives the reply, it uses the physical Ethernet address to send the Internet packet directly to the destined host.

2.4.6 Routing Information Protocol

The **Routing Information Protocol (RIP)** enables gateways to broadcast their current routing database to host and networks that are connected directly to them.

The Connection implements the RIP protocol through its dynamic routing server (UCX$INET_ROUTING.EXE), which runs as a subprocess to the VMS Internet ACP.

2.4.6.1 How Dynamic Routing Works The dynamic routing server listens on a UDP socket for packets with routing information. If the host is a gateway (an internetwork router), it periodically supplies copies of its routing tables to any host or networks that are directly connected to it.

When the dynamic routing server is started, it reads the UCX$ROUTE.DAT (located in SYS$SYSTEM) file to create entries in the Internet routing table for gateways that were defined by the system manager. Then it finds all active Internet interfaces (except those marked in loopback). If multiple interfaces are present, it is assumed that the host forwards packets between networks. The dynamic routing server then transmits a RIP request packet on each interface. If the interface supports broadcast packets, the dynamic routing server sends a broadcast packet; otherwise, it sends a normal packet. Then the dynamic routing server listens for RIP request and RIP Response packets from other hosts.

When a RIP request packet is received, the dynamic routing server formulates a reply based on the information maintained in its internal tables. The RIP response packet it generates contains a list of known routes, each marked with a hop-count metric. The metric associated with each route returned provides a metric that is relative to the sender. A hop count of 16 or greater is considered infinite.

The RIP response packets received by the dynamic routing server are used to update the Internet routing tables, if any one of the following conditions is satisfied:

- No routing table entry exists for the destination network or host, and the metric indicates the destination is reachable. That is, the hop count is not infinite.
- The source host of the packet is the same as the gateway (router) in the existing routing table entry. That is, updated information is being received from the gateway (internetwork router) through which packets for the destination are being routed.
- The existing entry in the routing table has not been updated for some time (defined to be 90 seconds) and the route is at least as cost effective as the current route.
- The new route describes a shorter route to the destination than the one currently stored in the routing tables. To decide this, the metric of the new route is compared against the one stored in the gateway's internal routing tables.

When an update is applied, the dynamic routing server records the change in its internal tables and generates a RIP response packet to all hosts and networks to which it is directly connected. The dynamic routing server waits a short period of time (no more than 30 seconds) before modifying the Internet routing tables to allow possible unstable situations to be resolved.

In addition to processing incoming packets, the dynamic routing server periodically checks the Internet routing table entries. If an entry has not been updated for 3 minutes, the entry's metric is set to infinity and marked for deletion. Deletions are delayed an additional 60 seconds to ensure the invalidation is propagated throughout the network (Internet).

Hosts acting as gateways supply their routing tables to all directly connected hosts and networks every 30 seconds. The RIP response is sent to the broadcast address on networks capable of that function, to the destination address on point-to-point links, and to the gateway's own address on other networks. The normal routing tables are bypassed when sending RIP responses.

The reception of RIP responses on each network is used to determine if that network and interface are functioning correctly. If no RIP response is received on an interface, another route may be chosen to route around the interface, or the route may be dropped if no alternative is available.

2.4.7 File Transfer Protocol

The **File Transfer Protocol (FTP)** allows authorized users to log in to a remote host, identify themselves, list remote directories, copy files to or from the remote host, and execute a few simple commands remotely.

2.4.8 Telnet Protocol

The Telnet protocol enables you to access any system on your network running the Telnet server software. When you access Telnet, it establishes a virtual terminal connection between your terminal and the specified host. Once a connection is established with a remote host, it appears as if your terminal is connected directly to that host. For more information see the *VMS/ULTRIX Connection User's Guide*.

2.5 Berkeley Internet Name Domain Resolver

The Berkeley Internet Name Domain (BIND) service is a host name and address lookup service for the Internet network. The BIND service is implemented in a client-server model. The client software is referred to as the resolver. The resolver allows client systems to obtain host names and addresses from servers rather than from locally hosted databases. Therefore, you can use the BIND service to supplement the host address mapping provided by the local UCX$HOST file.

The BIND service breaks the Internet into a hierarchy of domains, similar to a tree structure. Each domain is given a label. The name of the domain is the concatenation of all the labels of the domains, from the root to the current domain, listed from right to left and separated by dots (periods).

A label must be unique within its domain. The entire BIND Internet hierarchy is partitioned into several zones, each starting at a domain and extending down to the leaf domains (individual host names or to domains where other zones start).

A zone is a subdivision of a domain and is a discrete, non-overlapping entity. Each zone is an area of authority for which a master server is responsible. The Network Information Center (NIC) maintains the zone files of the root domain BIND server and the seven top-level domains for the United States, which are listed in Table 2–1. In addition to these, there are several top-level domains for individual countries. Contact the NIC for more information about them.

Table 2–1 Top-Level Domains

Domain	Description
arpa	For the Arpanet (which is gradually being phased out).
com	For commercial institutions.
edu	For educational institutions.
gov	For the government.
mil	For military organizations.
net	For network-type organizations such as network service centers, consortia, and information centers.
org	For miscellaneous organizations such as professional societies and similar non-profit organizations.

For example, Digital has been assigned the domain "dec.com". Within Digital, a domain administrator has assigned labels to different sites. For example, a fully qualified host name may be "bridge.nashua.dec.com".

Typically, each BIND domain has a domain administrator (DA), who is responsible for coordinating and managing the domain. The DA also controls the assignments of the host and domain names.

If the BIND resolver is running on your system, the Connection searches the local host database to resolve a host name or address. If it is unsuccessful, then the Connection queries the BIND servers. If you issue a UCX SHOW HOST * command, again the Connection queries the local host database and then the BIND servers.

You can access the BIND resolver by using the socket interface routines gethostbyname( ) and gethostbyaddr( ), or the UCX QIO programming interface.

For information on configuring and managing the BIND resolver, see Chapter 3.

2.6 Internet Addresses

For a local host to communicate with a remote host, it must know the Internet address of the remote host. The Internet address has a total of 32 bits (four octets) and is composed of two parts: the network number (including information on the network addressing scheme) and the host number.

The network part of the address must be the same for all the hosts connected to the same network, and no two networks can have the same network number if they are connected in any way.

No two hosts on the same network can have the same host number.

2.6.1 Address Notation

There are two types of notation for an Internet address.

The common notation uses four fields separated by periods to describe the 32 bits. Each field ranges from 0 to 255; for example, 98.0.2.65. The default values for the middle two fields are zero, so you can leave out these fields when their values are zero. For example, you can represent an Internet address of 88.0.0.70 as 88.70.

The alternate Internet address notation conveys the same information, but instead of specifying four fields (octets), it has two parts: one for the network information and one for the host information. The network part is a value from 0 to 255, as it is for the common notation. However, the host part is always made up of only one part.

Alternate notation is recognized by the second field being greater than 255. For example, you would represent the common notation for Internet address 128.0.2.20 as 128.532 in alternate notation. You arrive at the host part of the alternate notation (532) by the following calculation:

$$(256 * 2) + 20 = 532$$

As another example, Internet address 98.0.10.65 in common notation is represented as 98.2625 in alternate notation. The host part of the alternate notation (2625) is arrived at by the following calculation:

$$(256 * 10) + 65 = 2625$$

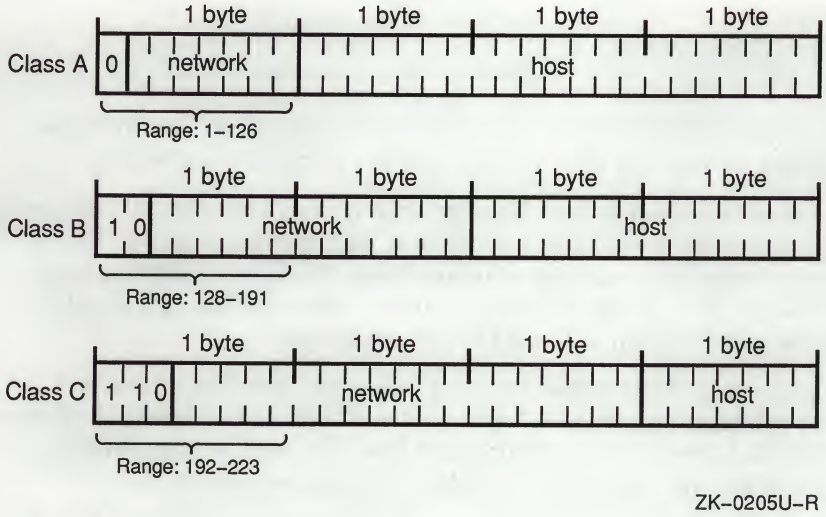
2.6.2 Network Classes

The network number provides two kinds of information: the network addressing scheme and the network number itself. Three types of network addressing schemes are supported: Class A, Class B, and Class C. The class type used depends on how the network is configured.

The four octets (fields) in the 32-bit Internet address are used in different ways to specify the class type, network number, and host number. The high-order bits in the network number designate the network class of the Internet address. For a Class A network the first high-order bit is 0. For a Class B network the first two high-order bits are 10. For a Class C network the first three high-order bits are 110.

Figure 2–3 shows the bit positions of the Internet address for the three network classes.

Figure 2–3 Internet Network Classes



For a Class A network, the first field specifies the network number and class, and the remaining three fields specify the host number (and a subnet address, if subnetworks are being used; see Section 2.7). The first field can be from 1 to 126, inclusive. By convention, 127 is reserved as the loopback address. Loopback is used for testing the connectivity to a specific host in the network.

For a Class B network, the first two fields specify the network number and class, and the remaining two fields specify the host number (and a subnet address, if subnet networks are being used). The first field can be from 128 to 191, and the second field can be from 1 to 254.

For a Class C network, the first three fields specify the network number and class, and the remaining field specifies the host number. The first field can be from 192 to 223, the second field can be from 0 to 255, and the third field can be from 1 to 254. Subnet routing is not generally used with a Class C network because there are 8 bits in the host field. Table 2–2 lists the ranges of the network numbers for the three network classes.

Table 2–2 Network Number Ranges

Class	Number
A	1–126
B	128.0–191.254
C	192.0.0–223.255.254

To determine which network class is best for a particular site, you need to know the number of hosts on the network and the number of networks.

The Class A network is best suited for sites with a few networks but numerous hosts, because it has 24 bits in the host part of its Internet address. The 24 bits allow for the most host-number combinations. There are only 7 usable bits in the network part of the Internet address, which leaves 126 usable network-number combinations (0 and 127 are reserved).

The Class B network is best suited for sites where the number of networks is about equal to the number of hosts, because the 32 bits of the Internet address are evenly divided between the network and the host part of the address. There are 16 bits for the network and 16 bits for the host part.

The Class C network is best suited for sites with numerous networks but few hosts, because the network part of its Internet address has 21 usable bits. The 21 bits allow up to 2,097,152 network-number combinations, while the 8 bits of the host part of the Internet address can have only up to 254 host-number combinations.

If you are planning to set up a local area network, obtain a registered Internet address. This way, if you choose to connect your network with another network, you will not have to change your Internet addresses.

2.6.3 Network Mask

Subnet routing requires a different interpretation of the Internet addresses. One or two octets are taken from the host part of the address and used to specify subnetwork information.

The network mask informs the system which bits of the Internet address to interpret as the network, subnetwork, and host addresses. A network mask is a 32-bit number. There is a one-to-one correspondence between the 32 bits in the network mask and the 32 bits in the Internet address. (Note that the terms network mask and subnet mask can be used interchangeably.)

For each bit in the network mask that is turned on (binary 1), the corresponding bit position in the Internet address is interpreted as part of the network and subnetwork address.

The decimal number 255 is 11111111 in binary notation. The value 255 means that an entire 8-bit field is turned on because each bit position is a 1. Generally, the entire 8-bit field is turned either on (255) or off (0). Values other than 255 or 0 can be used, but by using 255 or 0 you make it easier for users to differentiate between the network, host, and subnetwork fields.

If the network mask bit position is part of the host field and is turned on, the corresponding bit in the Internet address is interpreted as part of the subnetwork address. If the network mask bit position is part of the host field and is turned off, the corresponding bit in the Internet address is interpreted as part of the host address.

Each bit in the first (leftmost) field of the network mask must be turned on (decimal value of 255, binary value of 11111111), because the first field of the Internet address must always be interpreted as the network address regardless of whether there are subnetworks. If a bit in the first field of the network mask is turned off, part of the network field of the Internet address is interpreted as part of the host address. This may cause errors.

The second and third fields are usually either 255 or 0, depending on how the Internet address is to be interpreted. The fourth field is usually 0, to indicate it as the host address.

Figure 2–4 illustrates how different network masks affect the subnetwork address. As illustrated, a Class A network mask is usually 255.255.0.0 or 255.255.255.0. When the network mask is 255.255.0.0, the first byte is the network address, the second byte is the subnet address, and the third and fourth bytes are the host address. If the network mask is 255.255.255.0, the first byte is the network address, the second and third bytes are the subnet address, and the fourth byte is the host address.

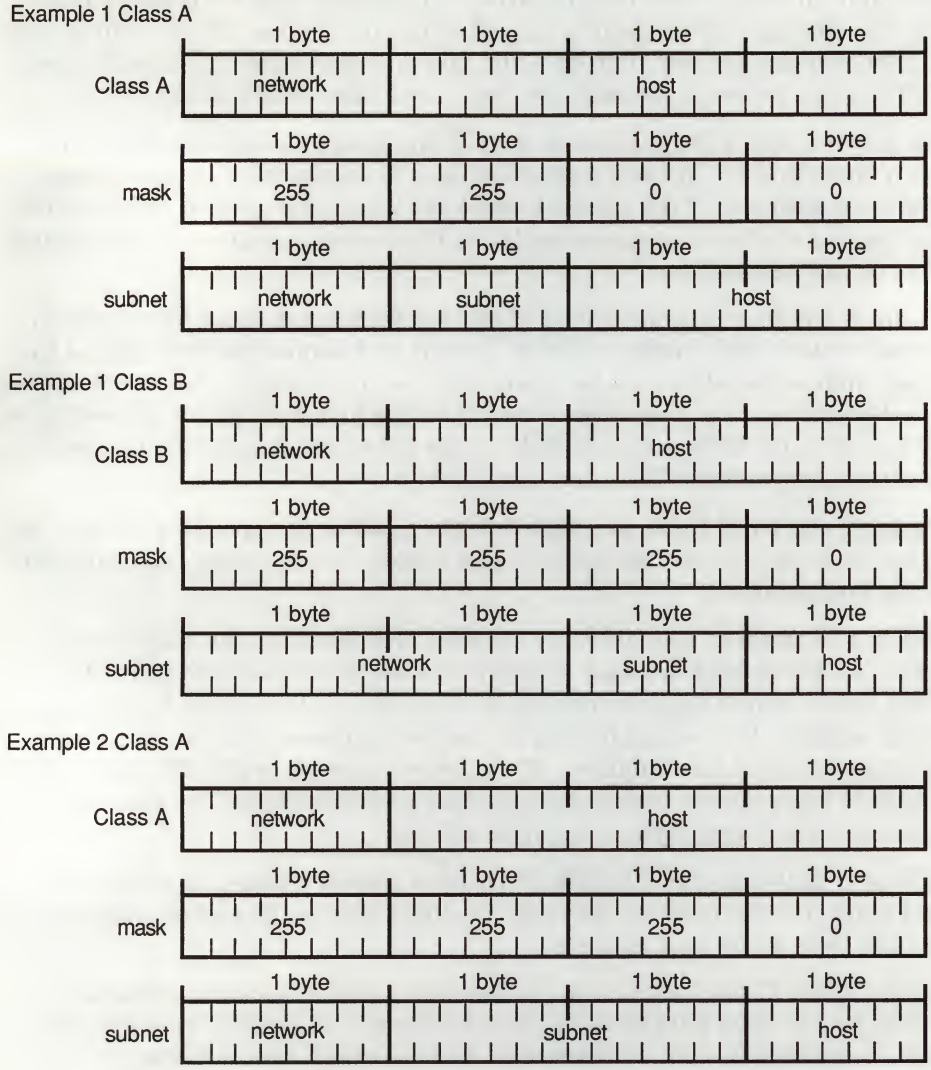
If a Class B network uses 255.255.255.0 for a network mask, the first and second bytes are the network address, the third byte is the subnet address, and the fourth byte is the host address.

Normally, Class C networks do not have subnetworks, because 8 bits are allocated for the host part of the Internet address. Eight bits may not be enough to divide between a subnetwork address and a host address.

The default network masks for each class are as follows:

- Class A — 255.0.0.0
- Class B — 255.255.0.0
- Class C — 255.255.255.0

Figure 2–4 Network Masks



2.6.4 Broadcast Mask

The broadcast mask interprets the Internet address as a broadcast address. The broadcast address allows messages to be sent to all the hosts on the network at the same time. If you use subnetworks, all the hosts on the same subnetwork must have the same Internet broadcast address.

The default format of the broadcast address consists of the network number followed by all 1s, but for compatibility it may be necessary to change the Internet broadcast address to the network number followed by all zeros, because some operating systems (UNIX BSD 4.2 and ULTRIX–32 prior to Version 1.2) require all zeros for a broadcast address. Problems can occur when systems using all zeros coexist on the same network as systems using all 1s. The two hosts may not be able to interpret each other's broadcast address.

The network number includes the subnet, if there is one.

If you know the Internet address and the network mask for a particular host, you can figure out the broadcast mask by using the following formula:

$$(NOT\ networkmask)\ OR\ (internetaddress)$$

For example, if a host has an Internet address of 128.50.100.100 and its network mask is 255.255.0.0 (the default), then its broadcast mask is 128.50.255.255. The *NOT* of its network mask is 0.0.255.255. You then substitute the first two fields of the Internet address for the two zeros to get the broadcast mask.

Table 2–3 lists examples of broadcast addresses.

Table 2–3 Broadcast Addresses

Host Internet Address	Host Number	Network Class	Network Number	Network Mask	Broadcast Address
3.0.0.10	10	A	3.	255.0.0.0	3.255.255.255 or 3.0.0.0
11.1.0.12	12	A	11.1.	255.255.0.0	11.1.255.255 or 11.1.0.0
129.39.0.15	15	B	129.39.	255.255.0.0	129.39.255.255 or 129.39.0.0
128.45.2.8	8	B	128.45.2.	255.255.255.0	128.45.2.255 or 128.45.2.0
192.0.1.8	8	C	192.0.1.	255.255.255.0	192.0.1.255 or 192.0.1.0
192.0.1.223	223	C	192.0.1.	255.255.255.0	192.0.1.255 or 192.0.1.0

2.7 Routing

A distinction is made between names, addresses, and routes. A host is given a name that is recognizable to users, such as HARRY or FOOBAR. The host name is associated with one or more Internet addresses. A route is the path over the network that information takes to get from one host to another.

The IP layer protocol deals primarily with addresses. It is the task of a higher protocol layer (for example, the application layer) to map the names to addresses. The IP layer maps the Internet addresses to local network addresses. It is the task of the local network or gateway to map from local network addresses to routes. Network routing is illustrated in Figure 2–1.

The following descriptions refer to Figure 2–5.

❶ The sending application program (application layer and host-to-host layer) prepares its data and calls on its Internet protocol layer.

The Internet protocol layer receives the data and the destination address as arguments of the call.

❷ The Internet protocol layer prepares a datagram header (iph1) which contains the destination Internet address and attaches the data to it. Knowing the Internet address of the destination host, the Internet protocol layer determines the network on which to send the data. If the destination host is on the same network as the local host, the destination Internet address is the address of the destination host. If the destination host is on another network, the destination Internet address is the address of the gateway that connects the local network to the destination network.

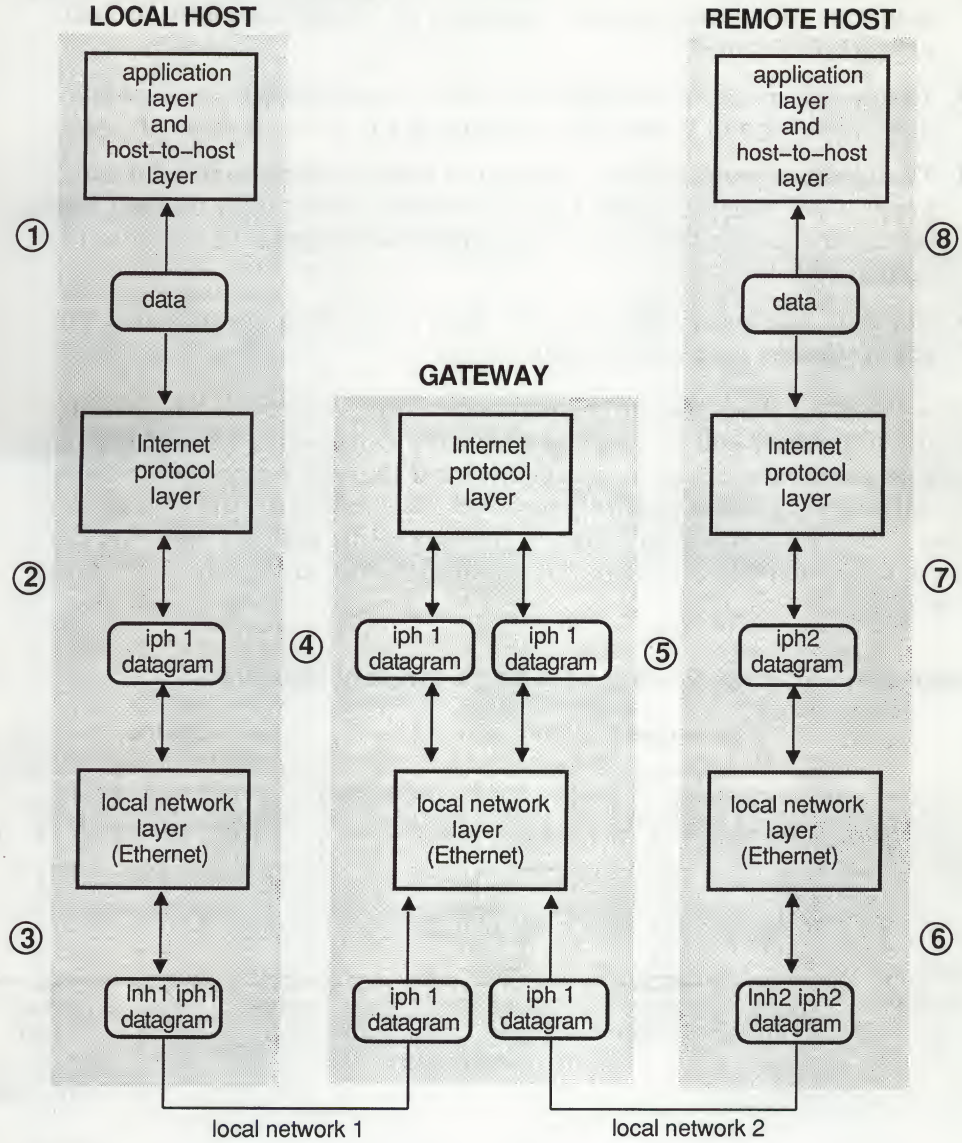
The Internet protocol layer sends this datagram to the network layer.

❸ The network layer creates a local network header (lnh1) and attaches the datagram to it. The datagram with the attached header is sent by means of the local network (local network 1).

❹ If the datagram is sent to a gateway host, the network layer of the gateway host removes the local network header (lnh1) and turns the datagram over to the Internet protocol layer.

❺ The Internet protocol layer determines the destination Internet address that the datagram is to be forwarded to from the Internet header (iph1). The Internet protocol layer determines a local network address for the destination host, and passes the datagram to the network layer for the network to send the datagram. The datagram contains a new Internet protocol header (iph2) that contains the destination Internet address.

Figure 2–5 Internet Routing



Legend

iph = Internet protocol header
lph = local network header

ZK–0090U–R

6. The network layer creates a local network header (lnh2), attaches the datagram to it, and sends the results to the destination host on local network 2.
7. The destination host removes the local network header (lnh2) at the network layer and passes the datagram to the Internet protocol layer.
8. The Internet protocol layer determines that the datagram is for an application program in this host. It removes the Internet protocol header (iph2), and passes the data to the application program in response to a system call.
9. The data, the source address, and other parameters are passed to the application as results of the call.

Subnetworking allows for organizing hosts within a network into logical groups. A network can be made up of several subnetworks. A host on another network can access a host on a subnetwork if there is a gateway connecting the networks, as illustrated in Figure 2–6. The data from the host on the other network is routed through the gateway to the network and onto the appropriate subnetwork, where the destination host ultimately receives the data.

Figure 2–6 Subnet Routing on a Single Physical Network

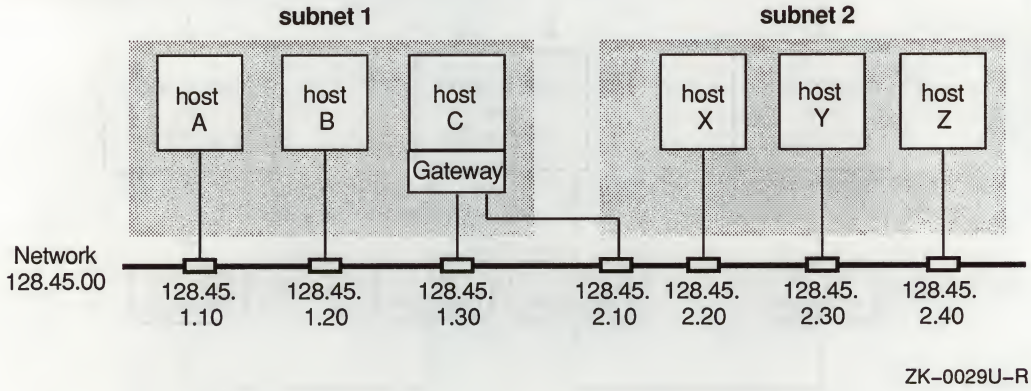
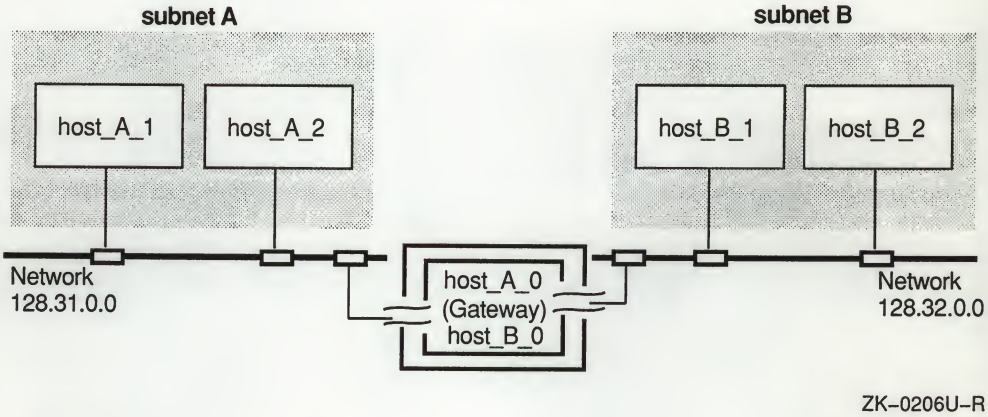


Figure 2–7 shows an example of subnet routing on multiple physical networks.

Figure 2–7 Subnet Routing on Multiple Physical Networks



2.8 Fragmentation

Fragmentation of an Internet datagram is necessary when the datagram originates in a local network that allows a large packet size and must transverse a local network that limits packets to a smaller size to reach its destination. Also, fragmentation is used when there may be no gateway, but applications send messages that are greater in length than the network layer supports. For example, the NFS server normally transfers information in 8000-byte packets, but the Ethernet network supports only 1518-byte packets. Therefore, the 8000-bytes datagram is fragmented into 6 datagrams that accommodate no more than 1518 bytes each.

A gateway can break up an Internet datagram into smaller Internet datagram fragments. The fragments can be further broken into smaller fragments at subsequent gateways.

The fragment format is designed so that the destination IP layer can reassemble fragments into datagrams.

3

Internet Management

After installing the VMS Internet software, you must configure your network and then start the Internet software. In a VAXcluster system, you need to install the software only once. However, you must configure and start the network software on each node in the cluster.

To set up the VMS system to communicate with other hosts on the Internet network, you must perform the following steps:

1. Install the Connection software.
2. Modify VMS SYSGEN parameters.
3. Configure the Internet interfaces.
4. Start the Internet software.
5. Set up the Internet databases.
6. Tune the network communications.

The *VMS / ULTRIX Connection Installation Guide* explains how to install the software. The following sections describe how to perform steps 2 through 6. This chapter also discusses the following topics:

- Creating the Internet interfaces
- Using a VAXcluster as a server
- Using the Address Resolution Protocol
- Displaying control and statistical information
- Shutting down Internet communications
- Troubleshooting the Connection Internet software

3.1 Modifying VMS SYSGEN Parameters

The following sections describe how to use the VMS System Generation Utility (SYSGEN) to change the system parameters that you need to modify for the Connection software.

3.1.1 Increasing the Nonpaged Pool Size

For the Connection to function properly, you need to increase the size of the nonpaged dynamic pool (controlled by the NPAGEDYN and NPAGEVIR parameters). NPAGEVIR defines the maximum size to which NPAGEDYN can be increased. Unless you are sure that NPAGEVIR is already large enough, you should add the same amount to both parameters. Digital recommends adding at least 342,000 bytes of nonpaged pool, which is the default maximum number of nonpaged pool needed by the Internet software.

To change these system parameters, perform the following steps:

1. Edit the SYS$SYSTEM:MODPARAMS.DAT file and add statements that increment the values of the SYSGEN parameters NPAGEDYN and NPAGEVIR. Edit the SYS$SYSTEM:MODPARAMS.DAT file as follows:

   ```
   ! add some nonpaged pool for the Connection software
   !
   ADD_NPAGEDYN=342000
   ADD_NPAGEVIR=342000
   ```

   You can use the following formula to calculate the amount of additional nonpaged pool your system requires:

   $$nonpaged\ pool = socket*1280+dbuf*1792+cbuf*256+mtusers*700+mrusers*700$$

   The symbols *socket*, *dbuf*, and *cbuf* indicate the maximum number of sockets, data buffers, and control buffers that you want to have on the system. (See Section 3.6.1.1 for a description of *socket*, *dbuf*, and *cbuf*.) The symbols *mtusers* and *mrusers* are the maximum number of Telnet and rlogin users, respectively.

   If you increase the cache size of the host in the future, Digital recommends that you increase the host's nonpaged pool size again.

2. Invoke the AUTOGEN command procedure to reset the parameter values.

3. Reboot the system to have the new parameter values take effect.

For additional information, see the VMS system management documentation.

3.1.2 Increasing Global Pages and Global Sections

For the Connection to function properly, you may need to increase the number of global pages and global sections (the SYSGEN parameters GBLPAGES and GBLSECTIONS). If the current settings for these parameters are inadequate, VMSINSTAL cannot start up the Connection software at the end of the installation procedure.

The Connection requires 1800 global pages and 27 global sections. To check the number of available global pages and global sections, use the WRITE command with the F$GETSYI lexical function:

```
$ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLPAGES")
4526
$ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLSECTS")
69
```

In this example, the F$GETSYI lexical function returns the number of available free global pages and free global sections and displays them on your terminal. If they are smaller than the required numbers, you must increase them.

To increase the global pages and global sections, edit the file SYS$SYSTEM:MODPARAMS.DAT and add the statements that increment the values of the SYSGEN parameters GBLPAGES and GBLSECTIONS. Then invoke the AUTOGEN command procedure to reset parameter values. Finally, reboot the system to have the new parameter values take effect. (For more information on the AUTOGEN.COM procedure, refer to VMS System Management documentation.)

Note *You must ensure that your system has the necessary global page and global section SYSGEN quotas for the installation. Failure to do so could cause the DCL tables to be corrupted in some situations.*

3.2 Configuring the Internet

The command procedure SYS$MANAGER:UCX$CONFIG.COM is used to configure the network environment for the local host. It also creates empty database files for the network and NFS components. You must execute this procedure on each VAXcluster member that will run NFS or the Internet software.

The following Internet-related files are created by this procedure:

- SYS$COMMON:[SYSEXE]UCX$HOST.DAT — Internet host database
- SYS$COMMON:[SYSEXE]UCX$NETWORK.DAT — Network database
- SYS$COMMON:[SYSEXE]UCX$ROUTE.DAT — Static route database

- SYS$SPECIFIC:[SYSEXE]UCX$INET_SET_INTERFACES.COM — Site-specific network startup procedure

The following NFS-related files are created by this procedure:

- SYS$COMMON:[SYSEXE]UCX$EXPORT.DAT — NFS export database
- SYS$COMMON:[SYSEXE]UCX$PROXY.DAT — NFS proxy database
- SYS$COMMON:[SYSEXE]UCX$NFS_SET_FS.COM — Site-specific NFS startup procedure

The host, network, export, proxy, and route databases are created in cluster-common directories and may be shared by all the nodes in the VAXcluster system.

All database files are created with a file protection that gives read access to the WORLD category. This protection allows nonprivileged users to read the database files, but it does not allow these users to modify the files.

Once the databases are created, define a logical name for each. Make the logical name system-wide, so the databases can be accessed by all users. This is done automatically for you by UCX$STARTUP.COM.

3.2.1 Executing the Procedure

Before executing the configuration procedure you must know the following information:

- The host names and Internet addresses for each Internet interface (Ethernet controller) on your system.
- If you use subnets, you must know the network mask.
- To configure and enable the BIND resolver, you must know the name of the BIND domain your system will reside in and the names and addresses of the BIND servers that will service your system.
- To enable dynamic routing, you must know whether you want your system to provide routing information to the network and whether you want it to supply the default route.

To invoke the procedure, log in to a privileged account and execute the following command procedure:

```
$ @SYS$MANAGER:UCX$CONFIG
```

The following privileges are needed to run the UCX$CONFIG.COM command procedure:

SYSPRV
SYSNAM
OPER

DETACH
BYPASS
CMKRNL
ALTPRI

If you lack the sufficient privileges to run the command procedure, the procedure aborts and you get an error message.

The SET INTERFACE commands generated by UCX$CONFIG contain options that you may want to modify when you tune your network. For example, each interface has ARP enabled by default.

Prior to running the command procedure, obtain an Internet address for each host you want to include in your network. You can obtain this information from your Internet network manager.

If your local area network does not have a registered Internet address, you must obtain one. In the United States you can obtain an Internet address by either:

- Calling the Network Information Center at 800–235–3155
- Sending electronic mail to hostmaster@nicddn.mil

Outside the United States, you should contact the following:

SRI International DDN — Network Information Center
Roon EJ217
333 Ravenswood Avenue
Menlo Park, California 94025
U.S.A.

3.2.2 Configuring the Internet Parameters

The UCX$CONFIG.COM command procedure performs the following steps:

1. Creates empty UCX$ROUTE and UCX$NETWORK databases and creates the UCX$HOST database with a LOCALHOST entry. If these files already exist, no action is taken.

2. Lists the Ethernet controllers that exist on your system. If it cannot find any, it displays the following message:

   `No Ethernet device on system host_x`

   The symbol *host_x* is the name of your system.

   For each Internet interface (Ethernet controller) it finds, the command procedure displays the following message:

   `Start of configuration questions for Internet interface ddx.`
   `ddx is the ethernet device ddxvax:`

3 Prompts you for the host name for the Internet interface it found:

```
The Internet host name is case sensitive.
If you want the host name to be lowercase, you must enclose your
answer in quotes. For example:
"remote"
Enter host name:  "june"
```

If the host name already exists, the following message appears:

```
Hostname host_x (address = ''ADDRESS') already exists in
host database.
```

The symbol *host_x* is the name that you just entered. You are prompted to enter another name. If you do not want to define an interface for this Ethernet controller, press Return.

If the host is a multi-homed host (a host with more than one Internet address) or a member of a VAXcluster, the following information is slightly different. For more information on multi-homed hosts, see Section 3.2.2.1. For more information on VAXclusters, see Section 3.2.2.2.

4 Asks you to enter the rest of the information for this Internet interface (Ethernet device):

```
Enter host name: "june"
Enter Internet address for my_host: 120.110.4.8
Enter Internet network mask for my_host [255.0.0.0]:
Enter broadcast mask for my_host [120.255.255.255]:
```

Notice that there are defaults for the network and broadcast masks. If you do not want to accept the defaults, refer to Section 2.6 for more information on setting the broadcast and network masks.

5 Asks you to confirm your answers for the preceding questions before they are used to define the network:

```
The following parameters will be used to define the
Internet interface SE0:

            Host name:          june
            Internet address:   120.110.4.8
            Network mask:       255.0.0.0
            Broadcast mask:     120.255.255.255

Is the above correct? YES
```

If you answer NO you are prompted for all the information again.

6 If the system is a member of a VAXcluster, a configuration is created for the cluster aliases of each interface previously defined. The network and broadcast masks for these configurations default to those specified for the original interface.

The command procedure continues to find Ethernet devices and to repeat these questions for each device.

7 Asks if you want to enable the BIND resolver. Answering YES to this question, configures your system as a BIND client. Therefore, you must make certain that there is a BIND server on your network.

```
UCX supports the Berkeley Internet Name Domain (BIND) client.
BIND is a network service that enables clients to name
resources or objects and share information with other objects
on the network.

Before configuring your system as a  BIND Client, you should
first be sure that there is at least one system on the network
configured as either a BIND primary or secondary server for
this domain.

Do you want to enable the BIND resolver [YES] ? YES
```

If you answer YES to this question, you must provide a domain name. If you answer NO to this question, you are not queried for a domain name; therefore, you should proceed to step 10.

8 Asks you to provide the name of the BIND domain for your local environment.

```
 WARNING - BIND domain and server names are case sensitive.
Lowercase names must be enclosed in quotes.

You must know the BIND domain name on which your system
will be a BIND client, in order to configure BIND. This
information should be available from your Internet network
adminstrator.

Enter the default domain name: "xyz.abc.com"
```

9 Asks you to provide the name of the remote BIND server for your client. You can use either the server's name or its Internet address.

When you have entered all the servers you want, enter a carriage return at the prompt.

```
The BIND server may be specified by its address or name;
however, if specified by name, there must be an entry
for it in the UCX$HOST database.

       You will be asked one question for each server.  Enter a
       <CR> at the prompt to terminate the list.

Enter the remote BIND server : 128.23.23.175
Enter the remote BIND server : "servra"
Enter the remote BIND server :
```

When you enter the server, the command procedure checks to see if the server is listed in the local host database. If it is, the procedure asks you for the name of the next server.

If you entered the server by name and the server is not in the local host database, the procedure asks you for the server's Internet address as follows:

```
Enter remote BIND server: MYSRVR
        MYSRVR is not in the local host database.

Enter Internet address for MYSRVR: 130.180.12.12
```

If you entered the server by address and the server is not in the local host database, the procedure asks you for the server's name as follows:

```
Enter remote BIND server: 130.180.12.13

        130.180.12.13 is not in the local host database.
        If you want to enter the server in the local host
        database, enter the server name. Otherwise, enter <CR>.

Enter remote BIND server name []: YOSRVR
```

You can enter a number of servers.

10 Asks you if you want to enable dynamic routing. If you answer YES, this host receives dynamic routing information form other hosts on the network.

```
If you enable dynamic routing, this host will listen
for all dynamic routing information coming from other
hosts to update its internal routing tables.
It will also supply its own Internet addresses to
routing requests made from remote hosts.

Do you want to enable dynamic routing [YES] ? YES
```

If you answer NO to this question, no other questions are asked.

11 Asks if you want this host to supply its routing information to other hosts on the network.

```
If you enable the 'supply' option of dynamic routing,
this host will supply dynamic routing information to
other hosts on the network whether it is acting as an
internetwork gateway or not.

Do you want this host to supply its dynamic routing information [NO]? YES
```

If you answer NO to this question, no further questions are asked.

12 Asks if you want this host to supply the default route. If you answer YES, this host will supply the default route.

```
        There can be one and only one host on your network
        which defines a default network route.

        If you want this host to supply the default network route,
        then all packets which cannot be routed to their
        destination host or network will be routed via this host.

Do you want this host to supply the default network route [NO] ? NO
```

Unless this host is going to be dedicated to routing, it is suggested that you answer NO to this question.

When the configuration process completes, the following message is displayed:

```
The Internet parameters have been defined successfully in the
command file SYS$MANAGER:UCX$INET_SET_INTERFACES.COM
```

Now you can start the Internet software or run IVP.

3.2.2.1 Defining Multi-Homed Hosts Multi-homed hosts are hosts that have more than one Internet address.

When you are defining the Internet interfaces, you usually specify them by their host names; this is not true for multi-homed hosts. When you define Internet interfaces for multi-homed hosts, you define them by their Internet addresses. When you connect to a multi-homed host, the Connection utilities try each address until a connection is successful. The addresses are tried in the order they were entered in the host database.

Additionally, if the Internet address for the host is modified, you must rerun the command procedure.

The following example discusses the differences you will encounter when defining interfaces for multi-homed hosts. In this example there are two Ethernet controllers, XEA0 and ETA0. The example assumes that the Internet addresses for "my_host" are already defined in the host database. Note that "my_host" is not an alias name and that no alias information is displayed.

```
Enter host name: "my_host"

 Host my_host exists in the database.
```

The command procedure informs you that "my_host" has multiple Internet addresses and lists the addresses. The addresses appear in the order in which they are defined in the host database.

```
 This host has multiple Internet addresses
 The Internet addresses for this host are:

                        150.100.4.8
                        150.110.4.9
```

Then the command procedure asks for the Internet address.

```
Enter Internet address [150.100.4.8]:
Enter Internet network mask for my_host [255.255.0.0]:
Enter broadcast mask for my_host [150.100.255.255]:
```

As with hosts with only one Internet address, the command procedure displays the information you provided and asks you for confirmation.

```
The following parameters will be used to define the
Internet device DE0:

 Host name:          my_host
 Internet address:   150.100.4.8
 Network mask:       255.255.0.0
 Broadcast mask:     150.100.255.255

Is the above correct? YES

End of configuration questions for Internet interface DE0
```

Because this host has more than one Ethernet controller, the command procedure now requests information on the second controller.

```
Start of configuration questions for Internet interface NI0.
NI0 is the ethernet device ETA0:

Enter host name: "my_host"
```

If you do not enter a host name at the prompt, the device is not configured.

The command procedure then confirms that "my_host" exists in the database and that it is a multi-homed host. This is because at least one of the host's Internet addresses was assigned to a previously defined interface.

Note that the command procedure flags the Internet addresses that have already been assigned to another controller and therefore cannot be used to define this controller. The command procedure also defaults to the next Internet address in the database.

```
Host my_host exists in the database.

This host has multiple Internet addresses

Addresses which have been assigned to another
interface are flagged with "*", and cannot be
assigned to this interface.

The Internet addresses for this host are:

                    * 150.100.4.8
                      150.110.4.9

Enter Internet address [150.110.4.9]:
Enter Internet network mask for my_host [255.255.0.0]:
Enter broadcast mask for my_host [150.110.255.255]:
```

As with hosts with single Internet addresses, the command procedure then displays the information and requests a confirmation.

```
The following parameters will be used to define the
Internet device NI0:

 Host name:          my_host
 Internet address:   150.110.4.9
 Network mask:       255.255.0.0
 Broadcast mask:     150.110.255.255
```

```
Is the above correct? YES

End of configuration questions for Internet interface NI0
```

The command procedure continues the processes until all the interfaces are defined for all the Ethernet controllers.

3.2.2.2 Defining Hosts in VAXclusters Defining hosts in VAXclusters is slightly different than for single hosts.

In the following example, because "my_host" was not put in quotation marks, the database access takes place on MY_HOST. The command procedure displays this information and then requests the Internet address, a network mask, and a broadcast mask.

This example shows that, instead of taking the default for the network mask, a subnet mask is to be defined. It also shows that, even though a subnet mask was defined, the command procedure provides the correct default for the broadcast mask.

```
 Start of configuration questions for Internet interface SE0.
 SE0 is the ethernet device ESA0:

Enter host name: my_host

 Host MY_HOST exists in the database.
 MY_HOST is an alias for my_host.

Enter Internet address [150.110.5.118]:
Enter Internet network mask for MY_HOST [255.255.0.0]: 255.255.252.0
Enter broadcast mask for MY_HOST [150.110.7.255]:
```

The command procedure then displays the information provided and requests a confirmation.

Following the confirmation, the command procedure displays the following information and asks if you want to define the VAXcluster:

```
 End of configuration questions for Internet interface SE0


 This CPU is a member of a VAXcluster.

 You may specify that the Internet interface(s) on this
 CPU will participate in an Internet VAXcluster.

 An Internet VAXcluster can be configured so  that  all
 cluster members appear to remote network  hosts  as  a
 single virtual host.

 The Internet VAXcluster is identified by  its  cluster
 host name and address, and must be activated  on  each
 of the hosts that  are members of the  Internet
 VAXcluster.

Define Internet VAXcluster [NO] ? YES
```

The command procedure then asks you for the name of the VAXcluster. Notice that the command procedure provides a default for the Internet address and that the network mask and broadcast mask for the cluster alias are defaulted to the masks defined for the interface.

```
 Start of cluster host questions for Internet interface SE0.
 SE0 is the ethernet device ESA0:

Enter CLUSTER host name: cluster

 CLUSTER Host CLUSTER exists in the database.
 CLUSTER is an alias for clustr.

Enter Internet address [150.110.5.120]:
Enter Internet network mask for CLUSTER [255.255.252.0]:
Enter broadcast mask for CLUSTER [150.110.7.255]:
```

3.3 Starting and Stopping the Internet Software

This section explains how to start and stop the Internet software. Starting and stopping the Internet software is controlled through the following command procedures located in SYS$MANAGER:

- UCX$INET_STARTUP.COM
- UCX$INET_SHUTDOWN.COM

Whenever you start or stop the Internet software, a message is sent to the operator's console stating that the Internet has been started or stopped.

3.3.1 Starting the Internet

You start the Internet network using the supplied command procedure UCX$INET_STARTUP.COM, located in SYS$MANAGER.

The Internet startup procedure invokes a site-specific command procedure UCX$INET_SET_INTERFACES.COM located in SYS$SPECIFIC:[SYSMGR]. Digital recommends that you make your site-specific network changes to this file.

The Connection startup procedure, UCX$STARTUP.COM, located in SYS$MANAGER automatically invokes the Internet startup command procedure.

3.3.2 Shutting Down the Internet

The Internet software is shut down with the UCX$INET_SHUTDOWN.COM procedure, which is called from the Connection shutdown command procedure, UCX$SHUTDOWN.COM. This procedure performs the following functions:

- Shuts down the communication with the network (STOP COMMUNICATION), which includes shutting down known services (Telnet and rlogin server, routing, and deleting all Internet interfaces).

- Deassigns the defined logical names.
- Deletes the installed images.

3.4 Setting Up the Internet Databases

There are three Internet databases (files) that need to be created, loaded, and maintained on an ongoing basis. Table 3–1 lists these Internet databases.

Table 3–1 Internet Databases

Database	Description
UCX$HOST.DAT	Allows users to refer to hosts by name rather than Internet address.
UCX$NETWORK.DAT	Allows users to refer to networks by name rather than network number.
UCX$ROUTE.DAT	Identifies any gateway hosts.

When these databases are created, they receive the default protection of the directory they are created in.

The following sections discuss these databases.

3.4.1 Host Database

The host database enables users to use host names. If the BIND resolver is enabled on your system, the host database must include the following information:

- The local host name
- The host names of your local Internet interfaces
- The host names of the BIND servers on the network

If BIND is not enabled on your system, you must enter the host names, Internet addresses for these hosts, and any alias names for these hosts into the UCX$HOST.DAT file.

After you have executed the UCX$CONFIG command procedure, the UCX$HOST.DAT file contains an entry for the local host, as well as a host database entry for each interface. (If you have a multi-homed host, you may not have a unique entry for each interface.) If you have configured the BIND resolver, the UCX$HOST.DAT file may also contain entries for BIND servers on the network.

UNIX host names are usually in lowercase, so you may want to create lowercase alias names for your uppercase host names.

You can use the SET HOST command to enter host names and aliases into your host file. Alternatively, if you already have an established network of UNIX hosts, you can use the CONVERT/VMS command to convert a host file formatted for UNIX to a host file formatted for VMS.

You can use the SET HOST command to populate your host database. For example, if you want to make an entry for a host named "harry" with an Internet address of 128.45.32.10 and alias name of BROWN, you would issue the following command:

```
UCX> SET HOST "harry" /ADDRESS=128.45.32.10 /ALIAS=BROWN
```

The following example shows you how you can populate your host file with entries from a host file formatted for ULTRIX. Use the FTP utility to copy the ULTRIX host file to your VMS system. You can indicate that the input host file is in Yellow Pages or BIND format by using the /YP qualifier.

```
$ FTP
FTP> connect "bridge"
220 bridge FTP server (Ultrix Version 4.9 Wed Sep 14 21:33:38 EDT 1988) ready.
Connected to bridge
FTP> login
_Username:: system_manager
331 Password required for system_manager
Password:
230 User system_manager logged in.
FTP> get etc.hosts etc.hosts
---> PORT 128,32,45,1,4,7
200 PORT command successful.
---> RETR ETC.HOSTS
150 Opening data connection for ETC.HOSTS (128.32.45.1,1031) (1637 bytes).
226 Transfer complete.
FTP> exit
221 Goodbye
$
$ UCX CONVERT/VMS HOST ETC.HOSTS/YP
$
```

3.4.2 Network Database

To allow users to refer to networks by name rather than by network numbers, you can enter the network names, Internet addresses for these networks, and any alias names for these networks into the UCX$NETWORK.DAT file. An empty network file is created when you execute UCX$CONFIG.

The easiest way to obtain the network names and Internet addresses for the file is to perform the following steps:

1 Obtain a copy of a */etc/networks* file from one of the UNIX hosts on your network.

2 Use the UCX CONVERT/VMS command to convert this file into a file formatted for VMS.

You can also add entries to the network database using the SET NETWORK command.

For example, to make an entry for network SPONGE, which has a network address of 128.44.33, you can issue the following command:

```
UCX> SET NETWORK SPONGE/ADDRESS=128.44.33.0
```

3.4.3 Route Database

There are two types of routes:

- Routes to hosts — used only when packets are sent to the remote host
- Routes to networks — used when packets are sent to any host in the remote network

The Connection supports both static and dynamic routing. Routes that are entered into the permanent on-disk route database are static. These routes can be supplemented by routes that are received dynamically by the dynamic routing server.

You can define routes into the permanent, on-disk database or volatile database by entering the following information with the SET ROUTE command:

- Destination host name or host address for routes to specific hosts
- Destination network name or network address for routes to all hosts on a network
- Gateway host name or gateway Internet address

When you use the SET ROUTE command, the routing information can be added either to the volatile or permanent database, depending on whether or not you use the /PERMANENT qualifier.

For example, if the network is active, the following command establishes a route to host "june" using SNAPPY as a gateway and places it in the volatile database:

```
UCX> SET ROUTE "june" /GATEWAY=SNAPPY
```

To add the same route to the permanent database, you would issue the same command with the /PERMANENT qualifier.

3.5 Using a VMS Host as Gateway Between Networks

To use a VMS host as a gateway between local area networks, your system must meet the following criteria:

- The system must have an Ethernet controller connected to each of the physical networks.
- Each of the Internet interfaces for the Ethernet controllers must be defined.
- The forward Internet communication parameter must be enabled with the UCX command SET COMMUNICATION/FORWARD.
- You start dynamic routing as a supplier with the UCX command START ROUTING/SUPPLIER.

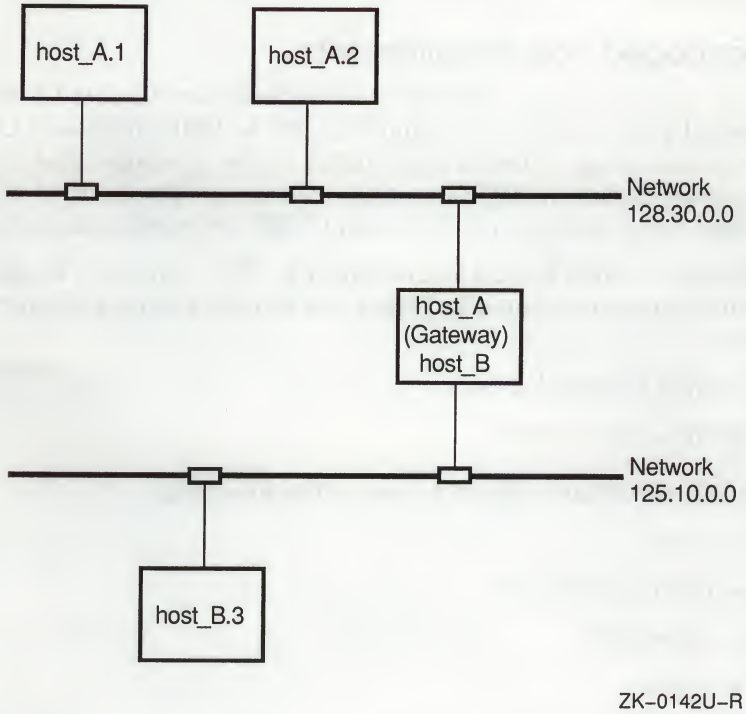
Under these conditions, when you start the Internet the VMS host automatically knows and routes packets from one local area network to the other and broadcasts its routing information to other hosts and networks it is connected to.

You can stop dynamic routing with the UCX command STOP ROUTING.

You can disable the gateway function by issuing the UCX command SET COMMUNICATIONS/NOFORWARD.

Figure 3–1 shows a sample network configuration with a gateway.

Figure 3–1 Sample Network with a Gateway



If dynamic routing is not enabled, you must define the routes on each of the hosts to be able to send packets through the gateway. For example, on HOST_B_3, you could define a route to network 128.30.0.0 as follows:

```
UCX> SET ROUTE 128.30.0.0/GATEWAY="HOST_B"
```

If you define the route in this manner, HOST_B_3 can send packets through the gateway to any host on network 128.30.0.0. If you defined the route as follows, you would only be able to send packets to HOST_A_1:

```
UCX> SET ROUTE "HOST_A_1"/GATEWAY="HOST_B"
```

To check if your routes are set up correctly, use the UCX command LOOP.

3.6 Tuning the System for Use with Internet

After the Internet software is started, you can use the SET COMMUNICATION command to change the VMS Internet parameters to improve the performance of the VMS system. (Note, you cannot change the minimum for the small and large buffers if the Internet has been started.) When you are satisfied with the Internet parameter settings for your system, edit these changes into the UCX$INET_SET_INTERFACES.COM procedure file (located in

SYS$SPECIFIC:[SYSMGR]). These Internet parameters are described in the following sections.

3.6.1 Nonpaged Pool Requirements

The Internet communication software uses buffer space allocated from the VMS nonpaged pool. Besides the buffers specific to VMS (IRPs and LRPs), the Internet communication software uses buffers in the nonpaged pool for its VMS Internet memory buffers (**MBUFs**) and *device-sockets*. The amount of required nonpaged pool space depends on the current VMS Internet implementation.

3.6.1.1 Memory Buffer Space Requirements There are two kinds of MBUFs, which are preallocated in larger blocks called control clusters and data clusters:

- Small buffers (control buffers)
- Large buffers (data buffers)

A small buffer stores data related to any of the following:

- Device-sockets
- Internal control structures
- Internet addresses
- Internet routes
- Internet packet headers

A small buffer is 256 bytes.

A large buffer is used for data buffering in the system space. A transmit data operation moves the data from the user process space to the large buffer before transmitting the data. A receive data operation stores the received data in the large buffers until a user process is ready to read the data. A data buffer is 1792 bytes.

A specified number of the MBUFs are preallocated statically from the VMS nonpaged pool during the Internet startup. If there is a need, more MBUFs can be dynamically allocated. The allocation or deallocation is always done in clusters.

You can use the SET COMMUNICATION command to specify the minimum number of control and data MBUFs that can be preallocated. For example:

```
UCX> SET COMMUNICATION/LARGE=(MIN=15, MAX=30)
UCX> SET COMMUNICATION/SMALL=(MIN=30, MAX=300)
```

This specified value is the number of small and large static buffers and is equal to one cluster.

The default values for the MBUFs are as follows:

- Small minimum — 30
- Small maximum — 330
- Large minimum — 10
- Large maximum — 100

The maximum value for the small minimum is 127. The maximum value for the large minimum is 15.

The default values are optimal for most configurations. Change these values only if necessary.

When the maximum is reached, depending on the type of request, the request is temporarily put in the Internet wait-for-resource state or the request is denied and aborted.

The number of dynamic MBUFs is equal to the maximum number of specified MBUFs minus the preallocated number of specified MBUFs. You can use the SHOW DEVICE command to monitor buffer usage. For example:

```
UCX> SHOW DEVICE
```

The Connection optimizes the MBUFs for your system; therefore, the numbers that appear for the small and large buffers may be different than what you specified.

3.6.1.2 Device-Socket Space Requirements A device-socket consists of an Internet interface (a VMS pseudodevice) and a socket. The socket itself is allocated from the control MBUF pool.

When a VMS process assigns a channel to the template Internet pseudodevice (UCX$DEVICE or BG0:), an Internet pseudodevice is created and a data structure is allocated for the Internet pseudodevice. Conversely, a process deassigns a channel to an Internet pseudodevice, the device is deleted and the data structure is deallocated. At any given time, a VMS process using Internet protocols for communication must have a channel assigned to an existing Internet pseudodevice.

Each device-socket requires 1280 bytes of nonpaged pool. The default value for the maximum number of device-sockets is 30, so the default maximum number of nonpaged bytes for the device-sockets is 38,400 bytes (30 * 1280 = 38400). Use the UCX conmmand SET COMMUNICATION to change the maximum number of device-sockets. For example:

```
UCX> SET COMMUNICATION/DEVICE_SOCKETS=40
```

Use the SHOW COMMUNICATION command to monitor device-socket usage. For example:

```
UCX> SHOW COMMUNICATION/FULL
```

3.6.2 Broadcast Messages

Many applications use broadcast messages during interprocess communications. By default, the Connection requires users to have special privileges to send broadcast messages. However, you can enable or disable privilege checking on broadcast messages with the /[NO]BROADCAST qualifier of the SET COMMUNICATION command.

The /BROADCAST qualifier disables the privilege checking on broadcast messages. The /NOBROADCAST qualifier enables the privilege checking.

3.6.3 Checksum Calculations

A checksum calculation is used to validate the data received across the Internet. A 16-bit one's complement of the one's complement sum of all 16 bit words in the header and text is provided as the checksum mechanism for UDP and TCP. A 16-bit one's complement of the one's complement sum of all 16-bit words in the header (no text) is also provided as the checksum mechanism for the IP protocol.

During the process of computing the checksum, the checksum field is replaced with all zeros.

For very large data packets, checksum calculations may require significant computing time. Hence, the checksum calculations may reduce the Internet performance. You can improve Internet performance by disabling the checksum calculation, but this can have a negative effect on the accuracy of the communications.

You can monitor the number of packets that were dropped because of bad checksum, with the UCX command SHOW PROTOCOL.

If the number of packets dropped because of bad checksum is low on your system, you can disable the checksum calculations with the SET COMMUNICATION/CHECKSUM command. For example, to disable checksum calculations for UDP, issue the following command:

```
UCX> SET COMMUNICATION /CHECKSUM=NOUDP
```

3.6.4 IP Reassembly Time

The IP reassembly time is the time in which IP datagram fragments have to be reassembled. If the IP reassembly time interval expires before the system receives all the fragments of the datagram, the fragments are discarded.

You can change the IP reassembly time with the SET COMMUNICATION command. For example, to change the IP reassembly time to 20 seconds, use the following command:

```
UCX> SET COMMUNICATION /REASSEMBLY_TIMER=20
```

If the time interval is too small, too many datagrams will be discarded; if the time interval is too large, it may have a negative effect on the Internet performance.

3.6.5 Socket Buffer Size for UDP and TCP

The Internet communication software stores the user data in the system space. To control network usage of nonpaged pool, you can change the send and receive buffer quotas for the UDP/IP and TCP/IP device-socket.

The Internet software discards network messages received for a device-socket whose receive buffer quota has been reached. If the send device quota is reached, the Internet software places a process in a resource wait. A process with the noresource wait characteristic will receive an I/O error on a write request that exceeds the send buffer quota. (See the VMS documentation for information on resource-wait queues.)

You determine the quota for the device-socket by specifying a socket buffer size quota for the transmit and receive operations of the UDP and TCP protocols. Table 3–2 lists the defaults and maximum limits for the quotas.

Table 3–2 Queue Size Quotas for Device-Sockets

Operation	Default	Maximum
UDP/IP Receive	9000 bytes	128K bytes
UDP/IP Transmit	9000 bytes	128K bytes
TCP/IP Receive	4096 bytes	64K bytes
TCP/IP Transmit	4096 bytes	64K bytes

You can specify the socket buffer size quotas with the SET COMMUNICATION command. For example, to specify a receive socket buffer quota of 8600 and a transmit socket buffer quota of 9700 for the TCP/IP, you would issue the following:

```
UCX> SET COMMUNICATION /TCP_QUOTA=(RECEIVE:8600,SEND:9700)
```

Increasing or decreasing the socket buffer size quotas has an impact on protocol performance.

3.6.6 I/O Request Packets

The VMS/ULTRIX Connection Internet driver interacts with the VMS Ethernet driver through I/O request packets (IRPs) that are allocated from VMS nonpaged pool. When the Internet traffic is heavy, the Internet driver may require a large number of request packets. To minimize the effect on the other users of the VMS system, the Internet driver provides a quota for these request packets. You can set this quota with the following command:

```
UCX> SET COMMUNICATION /IRP=MAXIMUM=n
```

The value *n* is the quota of request packets. The default value for MAXIMUM is 200 packets.

If the number of Internet request packets has reached the quota, one of two events occurs:

- The process that is being serviced by the Internet driver is put in a wait state (if resource wait mode is enabled for the VMS process).
- The I/O request of the process is aborted with an SS$_EXQUOTA return code (if resource wait mode is disabled for the VMS process).

To speed up the allocation of the Internet request packets, the Internet driver maintains a lookaside list or queue of free Internet request packets that have been preallocated from the VMS nonpaged pool. You can use the following command to set the maximum number of packets in the list:

```
UCX> SET COMMUNICATION /IRP=FREE=m
```

The value *m* is the maximum number of request packets. The default value for FREE is 20 packets.

You can set both the Internet request packet quota and the maximum number of packets in the free list with the following command:

```
UCX> SET COMMUNICATION /IRP=(MAXIMUM=n,FREE=m)
```

In general, the number of free IRPs should not exceed the quota of IRPs. Also, the value of the IRP quota should not exceed the number of IRPs that you defined with the VMS system parameter IRPCOUNT.

You can show both the Internet request packet quota and the maximum number of packets in the free list by issuing the following command:

```
UCX> SHOW COMMUNICATION
```

For more information on setting the number of I/O request packets, see the SET COMMUNICATION command in Chapter 5.

3.7 Creating the Internet Interfaces

The Internet interfaces provide physical communication to the network and are the lowest path of communications. The only interface type supported by the VMS Internet software is Ethernet.

The Internet software allows one local Internet interface that implements loopback and several Internet interfaces, one for each Ethernet controller. These Internet interfaces perform the packet channeling to and from the Ethernet. Each Internet interface is identified on a host by an interface name.

Table 3–3 lists the allowed interface names.

Table 3–3 Interface Names

Interface Names	Description
lo0	The local interface (loopback)
se0, se1,...	ESA, ESB,... (DESVA Ethernet controller)
qe0, qe1,...	XQA, XQB,... (DEQNA, DELQA, DEQTA Ethernet controller)
de0, de1,...	XEA, XEB,... (DEUNA, DELUA Ethernet controller)
ni0, ni1,...	ETA, ETB,... (DEBNA, DEBNE Ethernet controller)
ze0, ze1,...	EZA, EZB,... (SEGEC Ethernet controller)
xe0, xe1,...	EXA, EXB,... (DEMNA Ethernet controller)

The Internet interfaces are created and the Internet communications are started as part of the site-specific UCX$INET_SET_INTERFACES.COM procedure (located in SYS$SPECIFIC:[SYSMGR]). This file is created by the configuration command procedure SYS$MANAGER:UCX$CONFIG.COM.

3.7.1 Internet Pseudo-Interface

The Connection also supports the Internet pseudo-interface. This structure enables you to extend your subnet routing capabilities. Using the Internet pseudo-interface enables a VMS host to use one Ethernet controller and act as a gateway between subnets on the same physical network. It also enables a VMS host to use one Ethernet controller while being part of multiple subnets on the same physical network. The maximum number of subnets allowed by the Internet pseudo-interface is 65535.

An Internet pseudo-interface has a 3-character name, followed by a maximum of 5 digits for the pseudo-interface unit number. The maximum unit number is 65535. The first two characters in the pseudo-interface name are the same as the two characters in the Internet interface name (such as, qe, de, se,

and ni). The third character indentifies the Ethernet controller number that corresponds to the pseudo-interface, as follows:

- a is Ethernet controller 0
- b is Ethernet controller 1
- c is Ethernet controller 2 . . .

The unit number is used to differentiate interfaces that have the same name.

For example, if you used the Internet interface structure and your VMS system had two Ethernet controllers (XQA, and XQB), you could have had the following Internet interfaces:

- qe0
- qe1

If you used the Internet pseudo-interface structure, in addition to qe0 and qe1, you can have the following pseudo-interfaces, each with its own Internet address, network mask, and broadcast mask:

- qea0
- qea1 . . .
- qea65535
- qeb0
- qeb1 . . .
- qeb65535

Using the Internet pseudo-interface structure enables you to extend your subnet routing.

3.7.1.1 Extending Subnet Routing Subnet routing is one of the most widely used methods of allowing a single network address to span multiple physical networks. There are many situations in which you may want to use subnet routing.

For example, if your company has only one Internet address assigned to it and there is more than one physical network in your company. In this case, you can use local gateways and assign a subnet address to each local physical network. This way your company's network appears like one network to systems outside the company. However, to systems inside the company, it appears as many different networks.

Another example of using subnet routing is the opposite of the previous case. In this case, there is a single local physical network divided into multiple logical groups, based on some local organizational criteria, where each group acts as a separate subnetwork. For example, the physical network from one floor may be divided in multiple groups, each group representing a laboratory, or an office. The network may be arranged this way for the purpose of managing and monitoring the network. In this case, although the subnetworks are connected physically, you can connect them logically using gateways and subnet routing.

3.8 Using a VAXcluster System as a Server

A whole VAXcluster system or some of the hosts in a VAXcluster system can be represented by a VAXcluster alias. This alias appears to other hosts in the network as another host so a remote host can address the cluster as a single host or address any cluster member individually. The VAXcluster alias is implemented with the Address Resolution Protocol (ARP).

The NFS server is an example of a network service that can treat a VAXcluster system as a single host.

The task of answering incoming requests is switched among the cluster hosts at the end of each cluster time interval. The cluster name will not be switched away from a host if there are any active TCP/IP connections to the cluster host interface on that host.

You can set the cluster time interval in minutes with the SET COMMUNICATION command, as follows:

```
UCX> SET COMMUNICATION /CLUSTER_TIMER=30
```

If the time interval is specified as 0, the time interval value is infinite; the cluster alias is owned by one host until either the system is taken down or the Internet interface is deleted. Use this technique when you want to direct traffic to a single host, unless that host is unavailable.

You can specify a VAXcluster alias with the SET INTERFACE command. Use the command only at or after the other host interfaces are created. Change the parameters for the SET INTERFACE command only after deleting and re-creating an interface.

For example, to enable the VAXcluster alias MANY, issue the following command:

```
UCX> SET INTERFACE qe0 /CLUSTER=MANY /C_NETWORK=255.255.0.0-
_UCX> /C_BROADCAST=128.45.0.0
```

The /CLUSTER and /C_ADDRESS qualifiers are mutually exclusive; that is, only one of the parameters can be used at a time. The specified values for the cluster host name or the cluster Internet address must have been specified in the host database.

You can configure Internet clusters with the UCX$CONFIG.COM command procedure. For more information on configuration, see Section 3.2.

3.9 Using the Address Resolution Protocol

The Address Resolution Protocol (ARP) dynamically provides the Ethernet address for a specific Internet address.

You can statically input an Ethernet address for a specific host using the SET ARP command. This mapping can be done under one of the following situations:

- A host system that you want to communicate with does not support ARP; therefore, you statically map Internet addresses to Ethernet addresses.
- ARP is running and a change has been made to the network interface on a host. You need to flush the Internet-Ethernet mapping tables to allow the new address change to be made known to the system.

For example, to map the Ethernet address for host "harry", issue the following command:

```
UCX> SET ARP AA-02-04-05-06-07 "harry"
```

3.10 Displaying Control and Statistical Information

The UCX commands allow you to display control and statistical information at six different levels. Table Table 3–4 shows the six levels and the UCX command you would use to display the information.

Table 3–4 Displaying Control and Statistical Information

Level	UCX Command
Communication parameters	SHOW COMMUNICATION
Protocol statistics	SHOW PROTOCOL
Interface counters and flags	SHOW INTERFACE

(continued on next page)

Table 3–4 (Cont.) Displaying Control and Statistical Information

Level	UCX Command
Device-socket counters, quotas, flags, packets, ports, addresses	SHOW DEVICE
Buffer allocation and usage statistics	SHOW COMMUNICATION/MEMORY
Database information	SHOW HOST, SHOW ROUTE, SHOW NETWORK, and SHOW ARP

The following sections provide more information on some of these commands. For a complete description of the Show commands, see Chapter 5.

3.10.1 SHOW COMMUNICATION Command

The SHOW COMMUNICATION command displays Internet parameters for the local host that was specified by the SET COMMUNICATION command. This command has three qualifiers: /CONTINUOUS, /FULL, and /MEMORY.

The /CONTINUOUS qualifier enables you to select the screen update interval. The default is every 4 seconds, but you can specify a different interval in seconds. You can only use the /CONTINUOUS qualifier if you specify a specific interface.

The /FULL qualifier provides detailed information for each buffer type. You must also specify the /MEMORY qualifier to use this qualifier.

The /MEMORY qualifier displays list and statistics for each buffer type.

Example 3–1 shows sample output of the SHOW COMMUNICATION /MEMORY command.

Example 3–1 SHOW COMMUNICATION

```
UCX> SHOW COMMUNICATION/MEMORY
                                  MBUF Summary
                        Small_static  Large_static  Small_dynamic  Large_dynamic
 Total buffers                    30            10              0              0
 Free                             20            10              0              0
 Busy
  Data                             0             0              0              0
  Header                           0             0              0              0
  Socket                           2             0              0              0
  Prot. control                    4             0              0              0
  Route                            2             0              0              0
  Socket name                      0             0              0              0
  Fragment reassembly              0             0              0              0
  Access rights                    0             0              0              0
  IP address                       2             0              0              0
  Access control                   0             0              0              0
 Size of cluster                7936         18176              0              0

                         Large waits    Large drops   Small waits   Small drops
 Buffers                           0              0             0             0
            Current            Peak          Waits         Drops          Free
 IRPs             6               6              0             0             2
                   Small cluster  Large cluster          Non UCX buffer
 Free                              0              0             1
```

The following list explains some of the information provided by this display:

- Free buffers — The number of free buffers equals the number of total buffers minus the number of busy buffers.
- Small buffer — You can tune small buffer allocation with the UCX command SET COMMUNICATION/SMALL_BUFFER command. The total number of small buffers equals the number of Small_static and Small_dynamic combined. The static buffers are preallocated. If the demand for small buffers exceeds the total number of small buffers, the counters for Small waits and Small drops are incremented.
- Large buffer — You can tune data allocation with the UCX command SET COMMUNICATION/LARGE_BUFFER. If there are drops and waits, increase the maximum (dynamic) number of buffers with the UCX commands SET COMMUNICATION/LARGE_BUFFER or SET COMMUNICATION/SMALL_BUFFER, accordingly.
- Fragment reassembly buffers — If the number of fragment reassembly buffers is approaching the total number of buffers, lower the reassembly timer to discard the fragments sooner with the UCX command SET COMMUNICATION/REASSEMBLE_TIMER.
- IRP waits and drops — If the number of IRP waits and drops is high, increase the maximum number of IRPs with the UCX command SET COMMUNICATION/IRP.

- Free IRPs — If the number of free IRPs is approaching the maximum IRPs (see SHOW COMMUNICATION for maximum IRPs), there was probably a large peak demand. If the current number of IRPs is low, decrease the maximum number of IRPs with the UCX command SET COMMUNICATION/IRP.
- Minimum and maximum buffers — If the number of free buffers is approaching zero or there are many drops or waits, increase the maximum (dynamic) number of buffers with the UCX commands SET COMMUNICATION/LARGE_BUFFER or SET COMMUNICATION/SMALL_BUFFER.
- Peak demand — If there are many free buffers, and many drops or waits, this means that there is a high peak demand. If the number of drops or waits continues to increase, increase the maximum number of buffers. If the number of free buffers is approaching the total buffers, you may be able to decrease the maximum number of buffers. Use the UCX commands SET COMMUNICATION/LARGE_BUFFER or SET COMMUNICATION/SMALL_BUFFER for these adjustments.

3.10.2 SHOW PROTOCOL Command

The SHOW PROTOCOL command displays the protocol counters for the specified protocol. If no protocol is specified, counters for all the protocols are displayed. This command has one qualifier, /CONTINUOUS.

The /CONTINUOUS qualifier enables you to select the screen update interval. The default is every 4 seconds, but you can specify a different interval in seconds. You must specify a specific interface to use this qualifier.

Example 3–2 is an example of the SHOW PROTOCOL command. Because no protocol was specified, all active protocols are displayed.

Example 3–2 SHOW PROTOCOL Display

```
UCX> SHOW PROTOCOL
 IP
   Bad IP Version drops:                                        0
   Unknown IP broadcast address drops:                          0
   Total datagrams:                                       1050283
   Datagrams with bad header checksums:                         0
   Datagrams with size smaller than minimum:                    0
   Datagrams with data size < data length:                      0
   Datagrams with header length < data size:                    0
   Datagrams with data length < header length:                  0
   Datagrams forwarded:                                         0
   Fragments received:                                       1163
   Fragments dropped:                                           0
   Fragments timed out:                                         0
   Packets received for unreachable destination:           100498
   Packets forwarded on same net:                               0
   Fragments dropped from reassembly queue:                     0
   Fragments queued in reassembly queue:                     1163
   Datagrams dropped during reassembly:                         0
   Datagrams successfully reassembled:                        229
 TCP
   Connect initiated:          117     Connect accepted:          193
   Connect established:        284     Connect closed:            506
   Connect dropped:             14     Embry connect drop:         30
   Attempt rtt:             79347     Succeeded rtt:           79186
   XMT Delayed ACKs:        48767     Connect timeout:             1
   ReXMT timeout:              78     Persist timeout:            10
   Keepalive timeout:        7973     Keepalive probes:         7884
   Keepalive drops:             4     Total XMT packets:      144529
   XMT packets:             83559     XMT bytes:             9534924
   XMT packet reXMT:           70     XMT bytes reXMT:          4454
   XMT ACK only:            49038     XMT window probes:          40
   XMT URG only:                0     XMT wind update pack:    11499
   XMT CTRL packets:          323     Total RCV packets:      176330
   RCV packets:            106831     RCV bytes:            31524762
   RCV chksum error:            0     RCV bad offset:              0
   RCV too short:               0     RCV dup only pack:         403
   RCV dup only bytes:        401     RCV part dup pack:           0
   RCV part dup bytes:          0     RCV bad order pack:        195
   RCV bad order bytes:      3687     RCV pack after wind:         0
   RCV bytes after wind:        0     RCV pack after close:        3
   RCV window probes:          19     RCV dupl ACKs:           10768
   RCV ACK for unXMT:           0     RCV ACK packets:         79603
   RCV ACK bytes:         9513182     RCV wind update pack:      334
```

(continued on next page)

Example 3–2 (Cont.) SHOW PROTOCOL Display

```
UDP
  Bad header checksums:                                    0
  Incomplete headers:                                      0
  Bad data length fields:                                  0
  Socket buffer drops:                                     0
  Unknown broadcast port drops:                       751498
  Total received datagrams:                             6736
  Dropped:                                                 0
ICMP
  Calls to icmp_error:                                17298
  Errors not generated - old message too short:           0
  Errors not generaged - old message was icmp:            0
  Messages with bad code fields:                          0
  Messages with < minimum length:                         0
  Bad checksums:                                          0
  Messages with bad length:                               0
  Message responses generated:                           19
```

The following abbreviations are used for the TCP counters display:

- rtt — retries
- XMT — transmit
- reXMT — retransmit
- ACK — acknowledge
- URG — urgent
- RCV — receive
- CTRL — control
- chksum — checksum
- dup — duplicate
- Embry — embryonic connections (connections not yet established)
- wind — window

3.10.3 SHOW INTERFACE Command

The SHOW INTERFACE command displays information about a specific interface. If an interface name is not specified, information about all the interfaces on the host is displayed. This command has one qualifier, /CONTINUOUS.

The /CONTINUOUS qualifier enables you to select the screen update interval. The default is every 4 seconds, but you can specify a different interval in seconds. You must specify a specific interface to use this qualifier.

Example 3–3 is an example of the SHOW INTERFACE command.

Example 3–3 SHOW INTERFACE

```
UCX> SHOW INTERFACE lo0
 Interface: lo0
   IP_Addr: 127.0.0.1          NETWRK: 255.0.0.0        BRDCST:
                        Ethernet_Addr:
     Flags: UP LOOP
                                    RECEIVE          SEND
   Packets                                6           274
     IP packets                           0           274
     Broadcast IP packets                 0             0
     Trailer 1 IP packets                 0
     Trailer 2 IP packets                 0
     ARP packets                          0             0
     Broadcast ARP packets                6             0
   Drops
     IP packets                           0             0
     ARP packets                          0             0
   Errors
     Hardware                             0             0
     Software                             0             0
   Restarting attempts                    0
   Successful restarts                    0
```

Drops indicate the number of packets dropped. Drops are usually caused by insufficient resources.

Hardware errors are errors detected by VMS data link devices (such as Ethernet drivers). The Connection detects software errors while transmitting packets to or from data link drivers. These errors are usually caused by invalid protocol headers.

3.10.4 SHOW DEVICE_SOCKET Command

The SHOW DEVICE_SOCKET command displays socket information for the specified device. The device-socket is composed of two parts: the Internet interface and the socket. The SHOW DEVICE_SOCKET command uses the device name to identify the device-socket and obtains information about the socket (port number/protocol).

Example 3–4 is an example of the SHOW DEVICE_SOCKET command. Because a specific device is specified, only information about that device is diplayed.

Example 3–4 SHOW DEVICE_SOCKET Display

```
UCX> SHOW DEVICE_SOCKET BG45/FULL
 Device_socket: bg45                          LOCAL                REMOTE
                                   Port:        513                  1019
          Type: STREAM             Host:     lassue               roboot
                                   Service:  RLOGIN    Terminal: TNA9: -> VTA37:

                                                          RECEIVE        SEND
                                       Queued I/O               0            0
      Q0LEN          0                Socket buffer bytes      0            2
      QLEN           0                Socket buffer quota   1500         1500
      QLIMIT         0                Total buffer alloc       0         1632
      TIMEO          0                Total buffer limit    6000         6000
      ERROR          0                Number of XONs           0            2
      OOBMARK        0                Number of XOFFs          0            2
                                       I/O completed          240          366
                                       Bytes transferred      468        28363

 Options:  KEEPALIVE OOBINLINE
 State:    ISCONNECTED PRIV
 RCV Buff: SEL
 SND Buff: SEL
```

The following list provides descriptions for the counters in the first column:

- Q0LEN is the number of sockets that are about to be connected to the specified socket.
- QLEN is the number of sockets that have established a connection, but have not yet been accepted by the specified socket.
- QLIMIT is the number of sockets that can be connected to the specified socket.
- TIMEO is not used.
- OOBMARK is the out-of-band mark.

3.10.5 SHOW HOST Command

The SHOW HOST command displays information from the host database about the specified host, including alias names for the host. If BIND is enabled, the SHOW HOST command also displays information on the specified host from the BIND database. If you do not specify a host name, information about all the hosts is displayed.

The qualifiers for this command include: /DOMAIN, /[NO]LOCAL, /OUTPUT, and /SERVER.

The /DOMAIN qualifier specifies the domains to be used by the local host for the current SHOW HOST command.

The /[NO]LOCAL quailifer specifies whether information is displayed from either the local or the BIND database. By default, information is displayed from both the local and BIND databases. /LOCAL specifies that only information from the local database is displayed. /NOLOCAL specifies that only information from the BIND database is displayed.

The /OUTPUT qualifier enables you to direct the output of the SHOW HOST command to a file.

The /SERVER qualifier enables you specify the BIND servers for the local host. This definition is in effect only for the current SHOW HOST command.

You can use wildcards for the *host_name* parameter. However, if you use an asterisk to complete a host name, no BIND information is displayed.

Example 3–5 is an example of the SHOW HOST command. In this example, the /LOCAL qualifier is specified; therefore only the hosts in the local database are displayed.

Example 3–5 SHOW HOST Display

```
UCX> SHOW HOST/LOCAL

     LOCAL database

Host address     Host name

138.180.6.60     aa80z, AA80Z
138.180.4.1      abbss.zz3.ddd.com, abbss, ABBSS, ab, a
138.180.6.8      alibam, ALIBAM, alb
138.180.5.5      alLpin, ALLPIN, allpine.zz3.ddd.com
138.180.6.30     amfer, AMFER
138.180.6.2      ankles, ANKLES
138.180.6.73     auntie, AUNTIE, maitai
   .
   .
   .
138.180.4.200    zlepin, ZLEPIN
138.180.20.1     zooley, ZOOLEY, zoo
138.180.6.37     zxtra, ZXTRA
```

3.10.6 SHOW ARP Command

The SHOW ARP command displays information on Ethernet, Internet address, host name, and ARP status. If you do not specify a host name, all ARP information is displayed. This command has one qualifier, /ADDRESS.

The /ADDRESS qualifier enables you to select the host by Internet address as opposed to host name.

Example 3–6 is an example of the SHOW ARP command.

Example 3–6 SHOW ARP Display

```
UCX> SHOW ARP
                                ARP table entries

Ethernet            Internet        Host name           ARP status
AA-00-04-00-67-FA   150.110.4.191                       INUSE CMPL TRAIL
                    150.110.5.117   host_1              INUSE
AA-00-04-00-49-4E   150.110.5.109   host_2              INUSE CMPL
08-00-2B-03-1F-81   150.110.5.31    host_3              INUSE CMPL PERM
AA-00-04-00-6A-4E   150.110.4.2     host_4              INUSE CMPL PERM PUB
```

The following list explains the information displayed in the example:

- Entry 1 — This host is not entered in UCX$HOST database. It supports trailer headers.
- Entry 2 — This entry is incomplete, the remote host may be unreachable, or an answer has not yet been received to the ARP request (CMPL not present).
- Entry 4 — This entry is permanent. Entries 1, 2, and 3 will time out.
- Entry 5 — This entry is permanent and public. This host will respond to ARP requests for host_4. All other entries will not be propagated to remote hosts.

3.11 Managing Dynamic Routing

You enable and configure the dynamic routing server when you configure the network by running UCX$CONFIG.COM.

Once the dynamic routing server is enabled, you can manage the dynamic routing processes with the following UCX commands:

- SET ROUTE — Enables you to create entries in the permanent or volatile route database
- SHOW ROUTE — Displays routing information
- START ROUTING — Starts the dynamic routing program
- STOP ROUTING — Stops the dynamic routing program

When stopped, the dynamic routing program flushes the Internet tables of the dynamically created routes, leaving only those routes that belong to the host's Internet active interfaces.

When you start or stop the dynamic routing program, informational messages are written to the Network Operator console and log file.

For more information on these commands, see Chapter 5.

3.12 Managing the BIND Resolver

You have the option of enabling and configuring the BIND resolver when you run the UCX$CONFIG.COM command procedure. If BIND is enabled on your system, you can also modify the BIND domain name and the server information with the following qualifiers of the UCX SET NAME_SERVICE command:

- /[NO]DOMAIN — Defines the default domain. The domain name is appended to all host name references made from the local process.
- /[NO]SERVER — Specifies the servers known to the local host. The BIND request is sent to the specified server.

When you use these commands with the /SYSTEM qualifier, the new information takes effect system wide. Otherwise, the new information is only in effect for your process; this can be used for debugging or system management purposes, and it does not affect other users. The /SYSTEM qualifier requires SYSPRV or BYPASS privileges.

For more information on configuring BIND, see Section 3.2.

3.13 Managing FTP

This section provides the folowing information on managing FTP:

- Starting the FTP server
- Shutting down the FTP server
- Defining logicals names for FTP

3.13.1 Starting the FTP Server

The FTP server is started by SYS$MANAGER:UCX$STARTUP.COM. However, if you have a host with multiple Internet interfaces and you want to restrict the server to only accept connections from a single interface, you must first make the following edits to the UCX$FTPD_STARTUP.COM command procedure:

- Uncomment the following line in the command procedure:

  ```
  $ !DEFINE/SYSTEM UCX$FTP_HOST "HOSTNAME"
  ```

- Replace "HOSTNAME" with the name of the host on which you want the server to run.

Once you have edited the command procedure, you must execute the command procedure to start FTP.

The FTP server creates a log file, SYS$MANAGER:UCX$FTP.LOG, containing an abbreviated dialog from FTP sessions. You can get a more complete log by defining the logical name UCX$FTP_EXTLOG in the system logical name table before starting the server.

The FTP server also creates a log file (UCX$FTPSERVER.LOG) in the login directory everytime a client logs into the VMS FTP server.

3.13.2 Shutting Down the FTP Server

To shut down the FTP server, you must execute the UCX$FTPD_SHUTDOWN.COM command procedure.

3.13.3 Defining Logical Names for FTP

Table 3–5 lists and explanins the logical names used by FTP.

Table 3–5 FTP Logical Names

Logical Name	Function
UCX$FTP_EXTLOG	Enables extended logging of errors in the UCX$FTP.LOG file.
UCX$FTP_HOST	Defines the name of the host on which the server runs.
UCX$FTPD_IDLETIMEOUT	Sets the idle time for a control connection between a client and server. By default the idle time is set to 15 minutes. You can set this value to a time that is appropriate to your system.
UCX$FTP_LOWER_CASE	Specifies that directory listings are displayed in lowercase. This logical name can be set at the system or process level. This option also applies to the names of files transfered using wildcards with the GET command and *mget* operations.

(continued on next page)

Table 3–5 (Cont.) FTP Logical Names

Logical Name	Function
UCX$FTP_NO_VERSION	Specifies that the version number is removed from the file specification during transfers. This logical name can be set at the system or process level. This option also applies to the names of files transfered using wildcards with the GET command and *mget* operations.
UCX$FTP_WNDSIZ	Sets the window size of TCP send and receive buffer. This is set to 9K in the UCX$FTPD_STARTUP.COM file which is the default value. You can modify this value to one that is appropriate for your system. The maximum the Connection allows is 64K. Note that some Internet implementations allow only 32K.

3.14 Managing the Telnet and rlogin Server

The Telnet and rlogin server is automatically started when you execute the Connection startup procedure (UCX$STARTUP.COM).

In the VMS/ULTRIX Connection, the Telnet server is closely integrated with Internet and the VMS Terminal Driver. This integration provides the best performance during network and terminal I/O operations. Additionally, the Telnet server shares internal data and control buffers with Internet. Careful tuning of Telnet and Internet parameters can provide better use of Internet and VMS system resources and better performance.

This section discusses the following topics:

- Tuning Telnet server parameters
- Tuning Internet parameters for Telnet
- Monitoring Telnet
- Managing requests that cannot be satisfied

3.14.1 Tuning Telnet Server Parameters

You can modify the following Telnet server parameters to achieve better performance and use of system resources. These parameters are defined in UCX$REMOTE_TTY_STARTUP.COM (located in SYS$COMMON:[SYSMGR]). If you modify these parameters, you must stop and restart the Telnet server.

Maximum number of sessions

The Telnet server places a limit on how many Telnet sessions can be established (10 by default). You can set this limit by defining the system-wide logical name UCX$TELNET_MAX_SES with the DEFINE/SYSTEM command. For example:

```
$ DEFINE/SYSTEM UCX$TELNET_MAX_SES 8
```

Although there is no connection between data buffers and sessions, you should have at least one buffer available for each session.

Note *The Internet maximum number of device-sockets will be automatically increased by the Telnet maximum number of sessions when you start the Telnet server, and decreased by that number when you stop the Telnet server.*

Maximum number of data buffers in the free list

The Telnet server shares internal data buffers with Internet. To optimize data buffer allocation, the Telnet server keeps a list of free internal data buffers.

The default number of internal data buffers is 10. To change this value, define the system-wide logical name UCX$TELNET_FREE_BUF with the DCL DEFINE/SYSTEM command. For example:

```
$ DEFINE/SYSTEM UCX$TELNET_FREE_BUF 5
```

Maximum number of internal control buffers in the free list

The Telnet server shares internal control buffers with Internet. To optimize internal control buffer allocation, the Telnet server keeps a list of free internal control buffers.

The default number of internal control buffers is 4. To change this value, define the system-wide logical name UCX$TELNET_FREE_IRP with the DCL DEFINE/SYSTEM command. For example:

```
$ DEFINE/SYSTEM UCX$TELNET_FREE_IRP 3
```

Maximum number of unit control buffers in the free list

The Telnet server allocates a Telnet unit control block (UCB) for each remote login session. To optimize UCB allocation, the Telnet server keeps a list of free UCBs.

The default number of UCBs is 4; however, you should modify this to the average difference between the number of users that log in minus the number of users that log out. To change this value, define the system-wide logical name UCX$TELNET_FREE_IRP with the DCL DEFINE/SYSTEM command. For example:

```
$ DEFINE/SYSTEM UCX$TELNET_FREE_IRP 3
```

In case you set your VMS system to use virtual terminals, there is also a virtual terminal UCB allocated for each remote login session. For more information on setting up and using virtual terminals, see the *Guide to Setting Up a VMS System*.

Welcome messages

You can modify the welcome messages displayed by the Telnet and rlogin server by defining the following logical names:

- UCX$TELNET_MESSAGE
- UCX$RLOGIN_MESSAGE

For example:

```
$ DEFINE/SYSTEM UCX$TELNET_MESSAGE "UCX TELNET Server"
$ DEFINE/SYSTEM UCX$RLOGIN_MESSAGE "UCX RLOGIN Server"
```

Login/logout network operator console and log file messages

You can enable the displaying of login and logout messages to the network operator's console and log file by defining the following logical names to be equivalent to the character string TRUE:

- UCX$TELNET_LOG_MESSAGE
- UCX$RLOGIN_LOG_MESSAGE

For example:

```
$ DEFINE/SYSTEM UCX$TELNET_LOG_MESSAGE "TRUE"
$ DEFINE/SYSTEM UCX$RLOGIN_LOG_MESSAGE "TRUE"
```

3.14.2 Tuning Internet Parameters for Telnet

You can modify the following Internet parameters to achieve better performance and use of system resources.

Number of preallocated Internet buffers

Using Telnet on your system may significantly increase the incoming and outgoing Internet traffic and consequently the use of Internet small and large buffers.

Telnet keeps a list of free data buffers, which are taken from the Internet data buffer pool. To avoid affecting the Internet buffer allocation, you need to increase the number of minimum Internet small and large data buffers (statically preallocated buffers) by at least the maximum number of free data buffers that the Telnet server will hold in the free data buffer list. You do this by editing the SYS$MANAGER:UCX$INET_SET_INTERFACES.COM file and adding or uncommenting the following line:

```
$ SET COMMUNICATION/SMALL_BUFFERS=(MIN=60)/LARGE_BUFFERS=(MIN=15)
```

Maximum number of sockets

Using Telnet on your system may significantly increase the number of Internet device-sockets. The Telnet server uses one device-socket to listen for incoming connection requests, and one device-socket for each Telnet remote terminal session established. The Connection software makes this adjustment automatically; however, if this is insufficient for your system, you may want to increase the maximum number of device-sockets (Internet socket quota) that are allowed on your system. You do this by editing the SYS$MANAGER:UCX$INET_SET_INTERFACES.COM file and adding or uncommenting the following line:

```
$ SET COMMUNICATION/DEVICE_SOCKET=n
```

Internet ACP byte limit quota

The Internet device-sockets used by the Telnet server are allocated to the Internet ACP. Consequently, you must increase the Internet ACP process byte limit quota. You do this by modifying the RUN command qualifier BUFFER_LIMIT in UCX$INET_STARTUP.COM (located in UCX$MANAGER). The value for BUFFER_LIMIT should exceed the sum of the following:

1. The maximum number of incoming Telnet connections times 5000
2. Plus the maximum number of incoming rlogin connections times 5000
3. Plus the number of Internet interfaces times the sum of the following:
   - 8 times the LRPSIZE for IP
   - Plus 2 times 8 times the LRPSIZE for trailer
   - Plus 4 times the LRPSIZE for ARP

3.14.3 Monitoring Telnet

You can use the Internet management commands to monitor the Telnet server. You can display all Internet device-sockets being in use by the Telnet server with the following command:

```
UCX> SHOW DEVICE_SOCKET/SERVICE=TELNET
```

The following command also displays any Internet device-sockets on the local host and port, the remote host and port, and the I/O flags and counters:

```
UCX> SHOW DEVICE_SOCKET/FULL
```

Additionally, for Internet device-sockets being used by the Telnet server, this command displays the Telnet device being used for the Telnet remote terminal session, as well as the VMS virtual terminal, if one is being used.

3.14.4 Managing Requests That Cannot Be Satisfied

The Telnet server sends a message to the VMS Network Operator for each Telnet login request that cannot be satisfied because of insufficient local resources, such as the following:

- Maximum Telnet sessions exceeded
- Not enough VMS nonpaged pool for servicing a new Telnet connection
- Maximum VMS LOGINS limit exceeded

3.15 Troubleshooting the Connection Internet Software

When you cannot communicate with a remote server, try to isolate where the problem is occurring. You may first want to check the following:

- Establish that the two hosts are able to communicate. On VMS, you can use the LOOP command to test the connectivity between the two systems.
- If the LOOP command fails, recheck your physical network connections and your Internet parameters.
- Make sure the Internet network and broadcast masks are correct for your interface.
- Check that the host addresses are correct in the HOST database.

3.15.1 Troubleshooting BIND

If you are having trouble with the BIND resolver, it may be because of the following:

- The BIND resolver requires VMS Version 5.0 or higher and VMS/ULTRIX Connection Version 1.3 or higher. Make sure that you are running these correct versions of VMS and the Connection.
- BIND is disabled. Use the SHOW NAME_SERVICE command to determine if BIND is disabled. If it is, use the SET NAME_SERVICE /ENABLE command to enable BIND.
- The SYSTEM transport, domain, or server parameters are not defined. If the system parameters are not defined, use the SET NAME_SERVICE /SYSTEM command modify the system parameters.

  You will not be able to enable the BIND resolver unless the parameters are set correctly.

  The BIND parameters are managed by three sets of logical names. If the logical names were modified with using UCX$UCP, its possible that the logical names are defined incorrectly. Therefore, BIND may be enabled, but not all of the parameters are defined or they may be defined incorrectly.

Note that it is not possible to modify the BIND state and server parameters correctly through DCL; you must use the UCX SET NAME_SERVICE command.

- The Internet is not active. Use a UCX SHOW INTERFACE command to determine if the Internet is active. If the Internet is not active, you must start the Internet.
- The specified servers are not reachable. You can use a UCX LOOP command to each server to determine if the servers are reachable.
- The host name may be case sensitive. You cannot modify the BIND database; therefore, try placing the host name in quotation marks. For example, SHOW HOST "myname". Most BIND server host names are in lowercase.
- Standard VMS wildcarding does not work with the BIND database. You can use only an absolute wildcard (for example SHOW HOST *). You cannot use a wildcard to complete a host name. For example, if you entered the following commands, only information in the local host database would be displayed:

```
UCX> SHOW HOST NAM*
UCX> SHOW HOST *E
```

It is also possible that there are naming problems.

The BIND database is accessed by all UCX commands. If you have a situation where a command used to work and now it appears that the command is not working, it is possible that the command was relying on a host name in the local database and now the host name is being resolved through BIND. If there are problems resolving the name, the command may appear to hang. Therefore, for commands that use host names to define Internet parameters, define those host names in the local UCX$HOST database. This is a requirement for SET INTERFACE command.

4

Network File System Server

This chapter describes how you can manage the Connection Network File System (NFS) Server. It discusses the following topics:

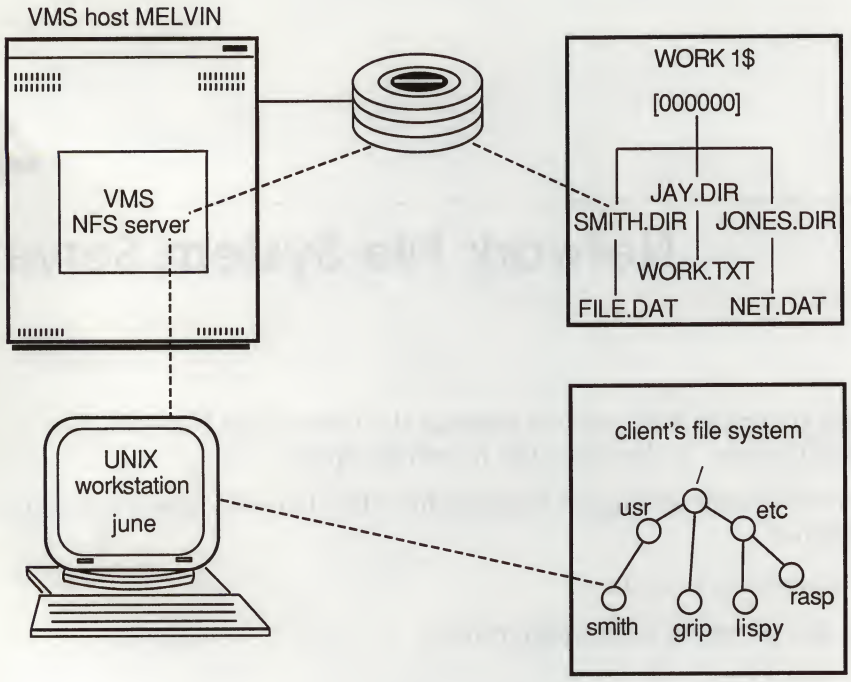
- The procedures for setting up accounts for NFS clients to access Connection file systems
- The maintenance procedures
- Tuning the server for better performance

4.1 Overview

The **Network File System (NFS)** server is the part of the VMS/ULTRIX Connection that allows NFS workstations to access data on a VMS host. The NFS server manager exports (makes available) directories to NFS clients. The NFS workstation user mounts remote directories located on the VMS NFS server onto local directories. The client then accesses files on the VMS server transparently through the local directory.

Figure 4–1 illustrates how a file system can appear to a client. The NFS server on host MELVIN exports the directory [SMITH] to the NFS workstation *june*. The user (usually root) mounts the directory WORK1$:[SMITH] onto the workstation's file system. The files in the VMS file system at directory [SMITH] appear to be in the NFS workstation directory */usr/smith*. If you issue a command to display the local directory */usr/smith*, you see the file *file.dat*.

Figure 4–1 A Client's File System



ZK–0143U–R

Figure 4–2 shows how the Connection NFS server interrelates with the other software components. The following sections describe the functions of these components.

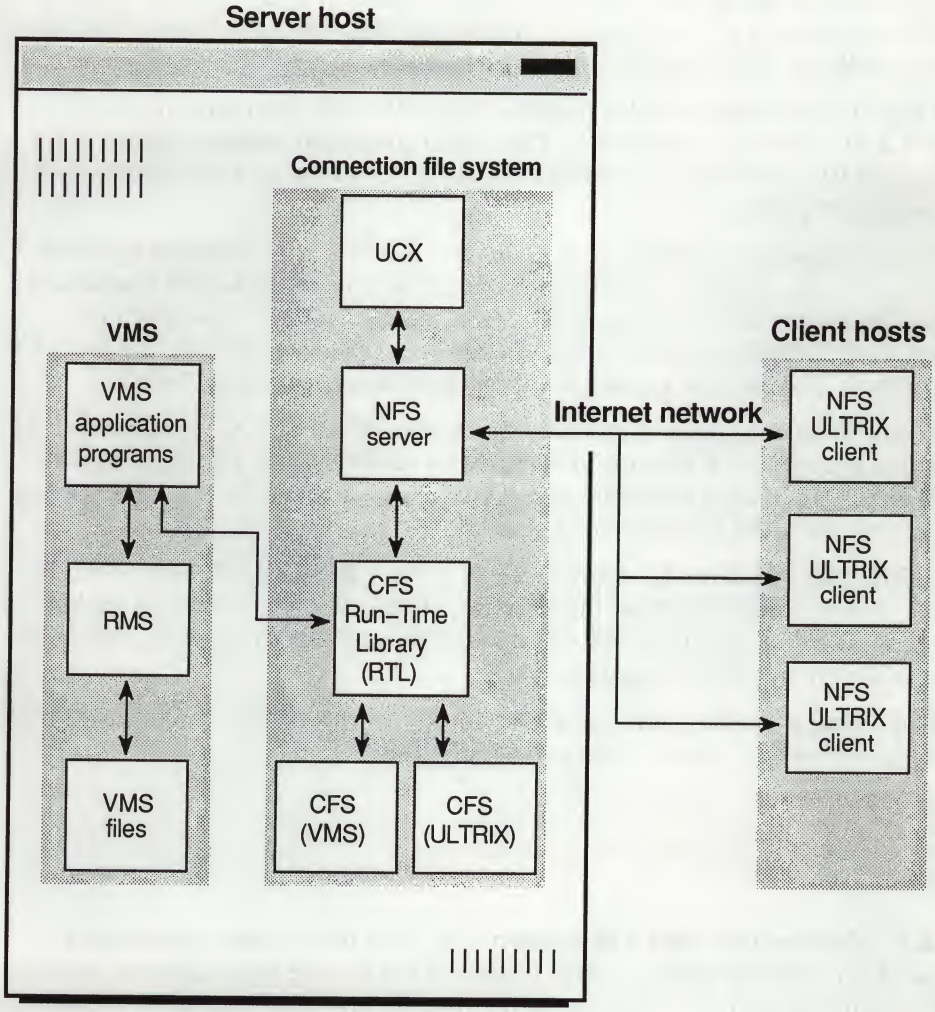
4.1.1 VMS and ULTRIX file System

There are several differences between VMS and ULTRIX file systems. Understanding these differences will help you understand the restrictions on the NFS client in accessing VMS files on the VMS server. It will also help you select the correct file system configuration for your users. If you are not familiar with these differences, see the *VMS/ULTRIX Connection User's Guide*.

4.1.2 Connection Management Utility

The **Connection Management Utility (UCX)** is the image that allows you to manage the Connection NFS Server. The UCX commands allow you to manage and tune the performance of the Connection NFS server.

Figure 4–2 Connection NFS Software Components



4.1.3 NFS Server

The **NFS server** (SYS$SYSTEM:UCX$SERVER_NFS.EXE) is a VMS image. It is multithreaded, meaning that it simultaneously processes multiple service requests from several NFS clients. The services that the NFS server provides are MOUNT, NFS, and PORTMAPPER.

4.1.4 Connection File System

The Connection file system is a collection of ULTRIX and VMS file systems that are organized as a single-level hierarchy. You build the Connection file system with the BIND and UNBIND commands.

The BIND command is analogous to a VMS MOUNT command in that it makes a file structure accessible. The BIND command makes either a VMS or an ULTRIX file system accessible to the NFS server and the Connection management process.

When a file system is bound, it is given an ULTRIX-style name of the form "/*name*," where *name* is a character string that can be up to 255 characters long. The root of the file system is referenced by "/name." The root of a Connection VMS file system is the Master File Directory (MFD); the root of a Connection ULTRIX file system is its top-level directory, called "root."

Once you determine your client needs, you can edit the BIND commands into your site-specific NFS startup procedure, UCX$NFS_SET_FS.COM, located in SYS$MANAGER. This command procedure is invoked by the NFS startup procedure when the Connection is started.

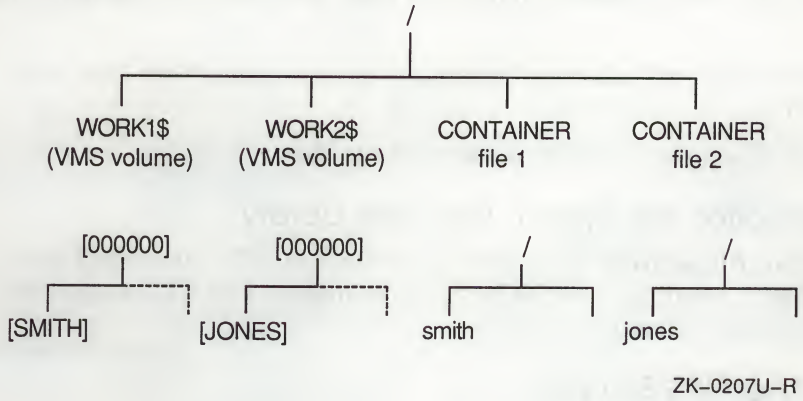
The UNBIND command is comparable to a VMS DISMOUNT command, in that it makes a file structure inaccessible. The UNBIND command makes either the VMS or the ULTRIX file system inaccessible to the NFS server or the management control program, UCX.

The following example shows how you would use the BIND commands to build the Connection file system shown in Figure 4–3.

```
UCX> BIND WORK1$: "/work1"
UCX> BIND WORK2$: "/work2"
UCX> BIND WORK1$:[GROUPA] "/groupa@work1"
UCX> BIND WORK2$:[GROUPB] "/groupb@work2"
```

4.1.4.1 Connection VMS File System A VMS file system resides on a Files–11 On-Disk Structure (ODS-2) disk. VMS Record Management Services (RMS) and the data management subsystem on the VMS operating system define the rules that govern the internal organization and the methods of accessing file data. VMS RMS and ODS-2 define a set of rules that govern files within the VMS file system. These rules define how files are named and cataloged within directories. For more information on VMS files and restrictions that might apply, see the *VMS/ULTRIX Connection User's Guide*

Figure 4–3 Connection File System



4.1.4.2 Connection ULTRIX File System A Connection ULTRIX file system consists of a special file called a **container file** and a collection of subdirectories. You create the container file by issuing a CREATE CONTAINER command. This container file, which is an RMS data file, contains the file system parameters and directory structure for an ULTRIX file system.

Each ULTRIX regular file is stored as a separate RMS data file using a CFS-assigned file name. The directory data files within the container file contain the ULTRIX file names and a pointer to the corresponding Files–11 data file. There is a VMS directory file for each ULTRIX directory in the container file, and all files cataloged in an ULTRIX directory are cataloged in a corresponding VMS directory. However, the ULTRIX directory hierarchy is not duplicated in the VMS directory hierarchy.

Note *Digital recommends that no regular VMS files be placed in the container file directory.*

4.1.4.3 Selecting the Correct File System for an NFS Client To determine which type of Connection file system is appropriate for each client, you must understand how an NFS client will use the VMS server.

An NFS client may want to use a Connection ULTRIX file system for the following reasons:

- The situation does not require extensive file sharing between client systems.
- Applications require an ULTRIX name space (file names are case sensitive).
- Applications require symbolic or hard links.

An NFS client may want to use a Connection VMS file system for the following reasons:

- The situation requires extensive sharing of data between VMS host and UNIX clients.
- The situation requires multiple versions of the files.

4.1.5 Connection File System Run-Time Library

The **Connection file system** (CFS) Run-Time Library (RTL) is a VMS run-time library that is used by both the NFS server process and UCX to process files within the Connection file system.

4.2 Managing the NFS Server

During the post-installation process, the configuration procedure, SYS$MANAGER:UCX$CONFIG, creates three empty NFS database files:

- SYS$COMMON:[SYSEXE]UCX$EXPORT.DAT — the EXPORT database
- SYS$COMMON:[SYSEXE]UCX$PROXY.DAT — the PROXY database
- SYS$COMMON:[SYSMGR]UCX$NFS_SET_FS.COM — the site-specific NFS server startup procedure

These database files can be shared by all VAXcluster hosts running the NFS server. The files are created with a file protection allowing WORLD read access but denying WORLD write access.

Before clients can use the VMS server, you must perform the following tasks:

- Register every NFS client with an NFS proxy account.

  You manipulate the proxy database with the ADD PROXY, REMOVE PROXY, and SHOW PROXY commands. See Section 4.2.2 for a description of what you must consider while selecting proxy accounts.

- Define the Connection file systems to be used by NFS clients.

  You define either a VMS or an ULTRIX file system to be a Connection file system using the BIND command. The BIND command creates a table of active file systems in memory, so the BIND commands must be executed each time the server is booted. Digital recommends that you include the BIND commands in the site-specific NFS startup file, UCX$NFS_SET_FS.COM, located in SYS$MANAGER.

- Define the client mount points in the EXPORT database.

You manipulate the export database with the ADD EXPORT, REMOVE EXPORT, and SHOW EXPORT commands. See Section 4.2.1 for more information on defining the export database.

- Modify the necessary NFS parameters.
- Start the NFS server process.

  The NFS server process is started with the command procedure UCX$NFS_STARTUP, located in SYS$MANAGER, which is automatically invoked by the Connection at startup.

- Examine the NFS error log file if NFS clients experience problems, and take the necessary corrective action.

  See Section 4.11 for more information on troubleshooting procedures.

4.2.1 Export Database

The export database allows you to control which Connection file systems a client can mount. The mount service allows an NFS client to mount a directory onto a local directory.

Before an NFS client can mount a file system on the server, you must identify which file systems or directories that particular client may mount by creating and maintaining the export database. The post-installation configuration procedure, SYS$MANAGER:UCX$CONFIG.COM, creates an empty export file called SYS$COMMON:[SYSEXE]UCX$EXPORT.DAT. You can also create an empty export file with the CREATE EXPORT command.

The ADD EXPORT, REMOVE EXPORT, and SHOW EXPORT commands allow you to maintain the EXPORT database. For example, using the ADD EXPORT command, you can identify mount points for clients. A client can mount any directory at or below the export point (subject to access control rules).

You must define a logical name for the database to create the file in a directory other than your default directory. The logical name should be systemwide, so the database can be accessed by all users.

4.2.2 Proxy Database

The proxy database is used to map NFS clients' UIDs, GIDs, and host names to VMS accounts. Therefore, any NFS client accessing files through the Connection must be registered in the proxy database.

The proxy database is a VMS index file. It is defined by the logical name UCX$PROXY when you run the UCX$CONFIG.COM command procedure. You can modify and manage the proxy database with the following UCX commands:

- ADD PROXY
- REMOVE PROXY
- SHOW PROXY

Before performing a file operation, the NFS server must authorize a client's access to the target file. Each NFS message contains the UNIX identity of its client, consisting of a UID, GID, and a source host name. The UID is normally a unique value assigned to each user at a site. The GID value is normally shared by members of the same work group.

When you start the NFS server, it reads the proxy database and the SYSUAF.DAT file into memory and creates the volatile proxy database. NFS then uses this volatile databse for its operations.

When NFS receives a message, it translates the client's host name, UID, and GID to a VMS user name. NFS uses this proxy VMS identity as the basis for access control when the NFS client accesses an NFS-mounted file and to control information about the user, such as the disk quota. When an NFS client accesses a file in an ULTRIX file system on the server, normal ULTRIX permission checking is done. Once the file operation is completed, the NFS server sends a response to the client. If the NFS server cannot map a client's UID, GID and host name to a VMS identity, access to the file is rejected with an authentication error.

NFS clients can modify file attributes, such as the file's owner UID and GID. In processing the set attribute request, the server must map the UID and GID into a VMS account using the proxy database. Thus, an NFS client cannot simply set a file's UID and GID to any arbitrary value; this UID and GID must be registered in the proxy database.

In the ULTRIX file system, the owner's identity (UID, GID and host name) and file attributes are saved in the container file. Therefore, NFS knows the owner's UID and GID. In the VMS file sytem, there is no equivalent of the ULTRIX UID and GID. Therefore, the UID and GID are mapped to a UID/GID pair using information in the volatile proxy database. If the NFS server cannot do the translation, the UID and GID are returned as –2 (**nobody** for ULTRIX).

If the NFS server resides on more than one host in a VAXcluster system, the proxy database and the export database can be managed as a homogeneous VAXcluster system (one proxy file on the VAXcluster system) or a heterogeneous VAXcluster system (a different proxy data base on each host in the cluster).

4.2.2.1 Superuser (Root) Account

Superuser, or **root**, is the account name given to a UNIX user whose UID equals zero and who is privileged to do any operation on a client provided with superuser privileges. Workstation users can use the root account to do system management operations, such as mounting NFS file systems.

If you expect your workstation users to manage their systems in this fashion, you need to register UID=0 with a proxy account and enter a proxy account for UCX$NFS_UID. Having root access on a workstation does not guarantee that the workstation user has root access on the NFS server. The NFS server maps UID=0 requests to the user whose UID and GID are controlled by the logical names UCX$NFS_UID and UCX$NFS_GID.

As shipped, the NFS startup file sets these values to be UID=–2 and GID=–2. If you want to allow superuser access to the NFS server, you must add records to the proxy database that map the client's UID to 0 and GID to 1. You can then give the account whatever privileges you chose on the VMS system.

4.2.2.2 Client Registration

Each NFS client must be registered in the NFS proxy file with the ADD PROXY command. The post-installation configuration procedure, SYS$MANAGER:UCX$CONFIG.COM, creates an empty proxy file, SYS$COMMON:[SYSEXE]UCX$PROXY.DAT. You can also create an empty proxy file with the CREATE PROXY command.

In choosing VMS accounts for your NFS clients, consider the following:

- You can give the NFS clients who already have a VMS account an NFS proxy to that VMS account.
- You need to give a VMS account to the NFS clients who do not already have a VMS account. These accounts do not have to be unique for each user; several NFS clients can use the same VMS account.

  You must be careful to choose UICs that match the intended work style. If you have a group of clients in the same work group who will be sharing files, you should assign the clients UICs in the same group.

For example, you can use the following command to register a client in the proxy database:

```
UCX> ADD PROXY smith /UID=53 /GID=45 /HOST="june"
```

You can specify a list of hosts for which the UID and GID are valid, or you can specify that all hosts are valid using an asterisk (*), as shown in the following example:

```
UCX> ADD PROXY nfs_super /UID=0 /GID=1 /Host=*
```

4.3 Setting Up and Creating File Systems

To set up a Connection VMS file system for NFS clients, perform the following steps:

1. Define the VMS file system as part of the Connection file system. For example:

   ```
   UCX> BIND DUA0: "/work1"
   ```

2. Export the Connection file system or a directory within the file system that can be remotely mounted by NFS clients. For example:

   ```
   UCX> ADD EXPORT "/work1/smith" /HOST="whale"
   ```

After these two commands have been executed on the server, the NFS client can mount the directory DUA0:[SMITH] onto a local directory */mnt* with the following command:

```
% mount server_name:/work1/smith /mnt
```

To make the bind permanent, add the BIND command to SYS$MANAGER:UCX$NFS_SET_FS.COM.

It may be easier to export the file system root rather than exporting directories under that root. However, exporting the root may not provide enough security for your environment. Generally, to maintain a more secure environment, you would export the top-level directory that you want a client to be able to access.

4.4 Setting Up a Connection ULTRIX File System

To set up a Connection ULTRIX file system, you must perform the following steps:

1. Create the Connection ULTRIX file system.
2. Bind the VMS volume on which the ULTRIX file system resides.
3. Bind the ULTRIX file system to the Connection file system.
4. Export a directory within the Connection ULTRIX file system.

4.5 Creating a Connection ULTRIX File System

To create an ULTRIX file system on the VMS NFS server, perform the following steps:

1. Create an empty ULTRIX file system on a VMS volume.
2. Bind the VMS file system on which the container file has been created.
3. Bind the ULTRIX file system.
4. Create a top-level directory for each NFS client.
5. Export the directories in the Connection ULTRIX file system.

4.5.1 Creating an ULTRIX File System

An ULTRIX file system resides on a Files–11 formatted disk and is represented as a set of files. The file system parameters and directory structure are kept in a data file called a container file. Each ULTRIX regular file is stored as a separate data file.

Creating a container file is comparable to initializing a new disk with a VMS volume structure because it creates the structure enabling users to create files. The newly created ULTRIX file system has an empty root directory.

For example, to create an ULTRIX file system on WORK1$:[ULTRIX_CONTAINER], issue the following command:

```
UCX> CREATE CONTAINER WORK1$:[ULTRIX_CONTAINER]/OWNER=[1,4]-
_UCX> /USER_NAME=nfs_super
```

The /USER_NAME qualifier selects a record from the NFS proxy file. The VMS UIC of this proxy account becomes the owner of the container file and directory. The corresponding UID and GID of this proxy account become the owner of the internal root directory of the newly created ULTRIX file system. If you do not specify the /USER_NAME qualifier, the proxy record whose UID equals zero becomes the owner of the container. If there is no record for UID=0, the container is not created.

The number of ULTRIX file systems to create depends on how you want to manage your system. You should have at least one ULTRIX file system for each work group.

4.5.2 Binding the VMS File System

You must bind the underlying VMS volume before binding the ULTRIX file system so the NFS server and the Connection management process (UCX) can access the VMS volume. If you do not add any entries to the export file for this VMS file system, then no NFS client will be able to mount directories in this VMS file system.

To bind the VMS volume on which the ULTRIX file system resides, use the following command:

```
UCX> BIND WORK1$: "/work1"
```

4.5.3 Binding the ULTRIX File System

Define the newly created ULTRIX file system to be part of the Connection file system with the following command:

```
UCX> BIND WORK1$:[ULTRIX_CONTAINER] "/remote_files"
```

To make the bind permanent, include the command in the site-specific NFS startup file SYS$MANAGER:UCS$NFS_SET_FS.COM.

4.5.4 Creating a Top-Level Directory for Each ULTRIX NFS Client

The newly created file system has a root directory and some internal system files. To display the newly created ULTRIX file system directory, use the following command:

```
UCX> DIRECTORY "/remote_files"
```

You must now create top-level directories for each of the intended clients who will access the ULTRIX file system. In creating the directory, you must specify who owns the directory and its protection. If you skip this step, it is unlikely that clients can create files in this file system. The root directory, which is the only directory that has been created at this point, is owned by the UID you specified with the /USER_NAME qualifier, and the default mode for this directory only allows write access to the owner.

The protection for file access includes read (r), write (w), and execute (x). Directory files use the execute protection to allow searching of the directory.

The system recognizes three classes of users when interpreting file or directory access:

- Owner
- Group
- Other

Because there are three types of permissions and three classes of user, you can set a total of nine permissions. These nine permissions are usually written in this format:

```
rwxrwxrwx
```

The leftmost three letters indicate owner permissions; the middle three, group permissions; and the rightmost three, other permissions. A missing permission is indicated by a dash (–). The nine permissions, or protections, are collectively known as the *mode* when you create a directory or *root_mode* when you create the container file.

The permissions are set with an absolute value given as an octal number. Each permission in the group of nine is represented by a 1, and each lack of permission is represented by a zero. Therefore,
rwxrw-r– translates to 111110100 (binary), or 764 in octal notation.

For example, suppose you want to create the ULTRIX directory */remote_files/user1*. This proxy account contains SMITH, which has a GID of 75, a UID of 50, and the following permissions:

- Owner permission of read, write, and execute
- Group read and write
- Other with no permission

To create this directory, use the following command:

```
UCX> CREATE DIRECTORY /USER_NAME=SMITH /MODE=760 "/remote_file/user1"
```

4.5.5 Exporting a Mountable ULTRIX File System or Directory

Before a client can mount a directory in the ULTRIX file system, you must export the directory using the ADD EXPORT command. For example, to allow client *june* to access directories in the Connection file system */remote_files*, issue the following command:

```
UCX> ADD EXPORT "/remote_files" /HOST= "june"
```

After these commands have been executed on the server, the NFS client can mount the directory */remote_files/user1* onto a local directory */mnt* with the following command:

```
% mount server_name:/remote_files/user1 /mnt
```

4.6 Security Considerations

Each NFS message contains the identification of the requesting client. This identification consists of the following items:

- A client host name
- A user ID (UID)
- A group ID (GID)

The NFS server processes requests only from clients who are registered in the proxy database. If you are working in a trusted environment, you may not want to restrict which hosts a client may use. In this case, you can create your proxy accounts so that every UID, GID pair is accepted from any host. Otherwise, you should specify a host or list of hosts for each individual client.

The NFS server maps the incoming NFS client identity to a VMS account using the proxy file. The server uses the privileges of this account to check file permissions. Be careful about granting privileges such as BYPASS, SYSPRV, or READALL to proxy accounts. The NFS server is working in a trusted environment, and clients are expected to have been authenticated by their client operating system.

You can enable some additional security options for the server by modifying the logical name UCX$NFS_SECURITY in the NFS server startup file, UCX$NFS_STARTUP.COM located in SYS$MANAGER. The server reads this logical name when it is started and applies the following security features:

- Setting bit 0 (value = 1)

  The NFS server grants the identifier UCX$NFS_REMOTE to each client, which creates a class of users who can access files using NFS. To prevent NFS access to selected objects, you can set access control lists (ACLs) on objects to deny access to subjects holding the UCX$NFS_REMOTE identifier.

- Setting bit 1 (value = 2)

  Setting this bit disables user-level mount requests. If this bit is set, only root is able to mount file systems on the server.

- Setting bit 2 (value = 4)

  Setting this bit disables the server from accepting client requests from nonprivileged Internet ports. If all your NFS client implementations use privileged ports, then you should use this feature to prevent a user from masquerading as an NFS client implementation.

  You control which Connection file systems a client can mount through the use of the export database. A client can mount any directory at or below the export point, subject to access-control rules. Exporting specific directories to specific hosts provides more control than exporting the root of a file system (or the MFD in a VMS system) to all hosts.

- Setting bit 4 (value = 16)

  Setting this bit enables you to restrict users' access to the NFS server through the SYSUAF file.

  The VMS Authorize Utility enables you to restrict a user's network access during certain hours of the day with the /NETWORK and /ACCESS qualifiers. When you add a proxy record to the volatile proxy database, the NFS server reads the network access information from the SYSUAF file. However, the NFS server does not automatically receive any changes made to the SYSUAF. Therefore, you must make sure that the information in the NFS virtual database reflects the information in the SYSUAF file.

  If access is denied for a certain time of day, the following message is written to the error log file and the NFS client receives an AUTH_BADCRED error in the reply message, which is translated into a permission denied message during the mount operation.

  ```
  UCX$-W-NFS_ACCNOA,  Access to the VMS account XXXX is denied
  ```

  Verification of the security setup for each NFS message reduces performance of the Connection because it requires at least three system calls and some additional calculations for each message. If the account has no restrictions on the network mode of operations, no verification is done. If the verification is disabled, no additional processing occurs.

By default, bit 4 is set to 0, which enables this feature. To disable the feature, set bit 4 to 1.

Note *The Connection does not allow the root of the Connection file system to be exported.*

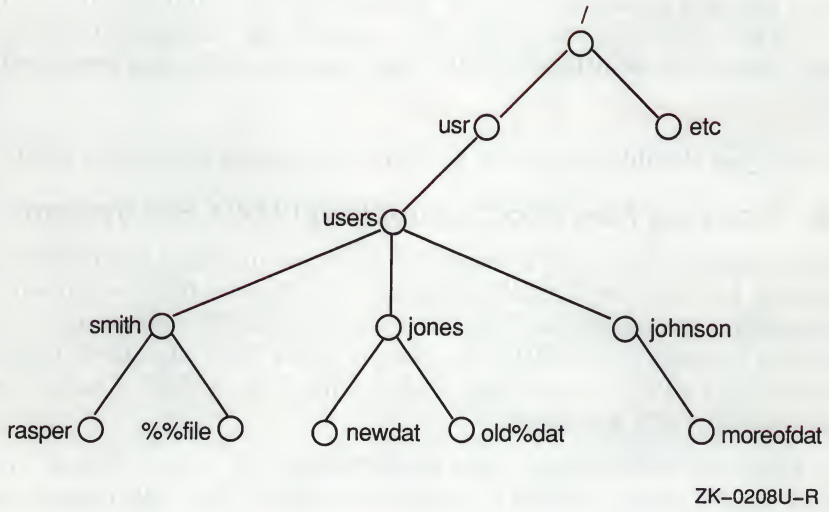
Consider that a Connection ULTRIX file system has been defined to be */groupa@whale*; this file system is shown in Figure 4–4. You can export the file system at the */usr* directory level by adding the following record to the export database:

```
UCX> ADD EXPORT "/groupa@whale/usr" /HOST=*
```

This record would allow clients on any host to mount any directory at or below */usr* in the file system. If you want to be more selective about which directories clients can mount, you can choose to export specific directories to specific clients, as shown by the following commands:

```
UCX> ADD EXPORT "groupa@whale/usr/smith" /HOST=S
UCX> ADD EXPORT "groupa@whale/usr/jones" /HOST=J
```

Figure 4–4 Exported File System



4.7 Maintaining the Connection ULTRIX File System

The Connection provides commands to maintain and examine Connection ULTRIX file systems on VMS. These commands allow you to perform the following tasks:

- Get directories
- Copy files
- Delete files
- Rebuild the UFS

4.7.1 Removing Links to a File

You can remove links to a file with the REMOVE FILE command. For example, to remove the link to a file named *letter* located at */usr/smith*, issue the following command:

```
UCX> REMOVE FILE "/usr/smith/letter"
```

4.7.2 Removing Links to a Directory

You can remove links to an ULTRIX directory with the REMOVE DIRECTORY command. For example, to remove the directory *smith* at */usr*, issue the following command:

```
UCX> REMOVE DIRECTORY "/usr/smith"
```

4.7.3 Deleting a Connection ULTRIX File System

You can delete a container file with all its directories and files by issuing the DELETE CONTAINER command. For example, to delete the ULTRIX file system created on WORK1$:[GROUP_A], issue the following command:

```
UCX> DELETE CONTAINER WORK1$:[GROUP_A]
```

Note that you should unbind the ULTRIX file system before you delete it.

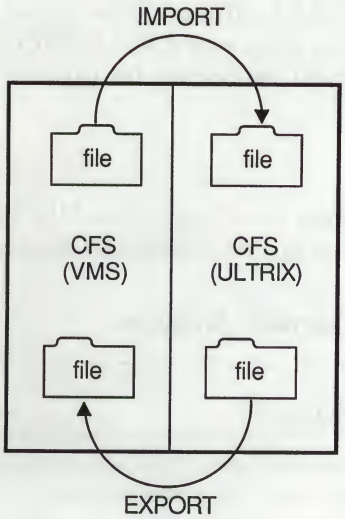
4.7.4 Copying Files into Connection ULTRIX File Systems

You cannot use a DCL COPY command to create files in a Connection ULTRIX file system because the ULTRIX directory structure is fully contained in the corresponding container file. You must use the EXPORT command to copy a file from a Connection ULTRIX file system into a VMS directory. Likewise, you must use the IMPORT command to copy a file from a VMS directory into a Connection ULTRIX file system.

If the VMS data file does not have the STREAM_LF record format, you may want to specify the /CONVERT qualifier to convert the VMS record structure to STREAM_LF.

Figure 4–5 illustrates the exporting and importing of files.

Figure 4–5 Exporting and Importing Files



ZK–0209U–R

4.7.5 Verifying the Integrity of a Connection ULTRIX File System

The Connection provides the ANALYZE CONTAINER command to check the integrity of a Connection ULTRIX file system. This command is similar in function to the DCL ANALYZE/DISK_STRUCTURE command.

For example, to verify the integrity of a Connection ULTRIX file system located in WORK1$:[GROUP_A], issue the following command:

```
UCX> ANALYZE CONTAINER WORK1$:[GROUP_A]
```

File system access to the container file is suspended while the container is being analyzed.

Note *The underlying VMS file system must be bound before you use the ANALYZE CONTAINER command.*

For a description of the ANALYZE CONTAINER command and its qualifiers, see Chapter 5.

You may want to check the Connection ULTRIX file system under the following circumstances:

- If you are having problems accessing a Connection ULTRIX file system. For example:
  - Disk read/write errors
  - Problems backing up a Connection ULTRIX file system

- If you are making copies of a Connection ULTRIX file system.

  The container file contains pointers to the locations of the actual files on the disk. If you copy the file system, you must run ANALYZE/CONTAINER after you copy the files so the pointers will be correct for the new location of the files.

- During system startup

Table 4–1 lists the important file components of a Connection ULTRIX file system that are normally verified by the ANALYZE CONTAINER command.

Table 4–1 Connection ULTRIX File System Components Analyzed

ULTRIX Item	VMS Conceptual Equivalent	Description
Super block	Home block	Contains the basic information on the internal structuring of the container file.
Inode	File header	Each file or directory has an inode that contains information which describes the file. The inode is a central definition of the file.
Directory	Directory	Contains the file names and directory hierarchy information. File name entries contain links to the inode information.
Bitmap	BITMAP.SYS	Contains the container file internal allocation information. Only one bitmap exists in the container file.

4.7.6 Backingup NFS-mounted Files

You can backup NFS-mounted files using standard VMS backup procedures. For more information, see the VMS documentation.

Users can continue to access a VMS file system during backup procedures without causing any problem. This is not true of an ULTRIX file system. Accessing files during a backup of a UFS can result in inconsistent container file information. Therefore, it is recommended that you unmount the ULTRIX file system before backup to prevent access.

4.7.7 Restoring NFS-mounted Files

The procedure for restoring NFS-mounted files depends on whether you are restoring a VMS file system or an ULTRIX file system. If you are restoring files from a VMS file system backup saveset, you can locate the files by name and restore the files. Restoring files from an ULTRIX file system backup saveset is more problematic because the VMS file names are not known. (How to restore files from an ULTRIX file system backup is explained in Section 4.7.7.1.

After you restore files, it is recommended that you run ANALYZE CONTAINER on the file system because the FID, times, and possibly the file sizes may be different.

4.7.7.1 Restoring ULTRIX File Systems The procedure for restoring files from an ULTRIX file system depends on whether you want to restore only a few files or many files.

Restoring a few files

If both the ULTRIX file system and the individual files exist, you can use the following steps to restore the file:

1. Obtain the inode number of the file you want to restore. You can get the inode number by issuing a UCX DIRECTORY/FULL command on the container file and the file name. For example, if the container file name was *mystuff* and the file was *myfile* you would issue the following command:

   ```
   UCX> DIRECTORY "/mystuff/subdirectory/myfile" /FULL
   ```

   The inode number is listed in the right upper corner in the File ID field as a decimal number.

2. Convert the inode number to hexadecimal. The hexadecimal number must be an eight digit number. If it is not, place leading zeros on the number to make it eight digits; then append $BFS. to the number. This gives you the VMS file name; for example, 00005503$BFS.

3. Perform steps 1 and 2 to find out the the VMS file name of the parent directory for the file. However, after you convert the inode number to hexadecimal append $BFS.DIR to the file name, so it will look as follows: 00012101$BFS.DIR.

   In this case the complete VMS path name for the *myfile* would be [MYSTUFF.00012101$BFS]00005503$BFS.

4. Once you have the complete VMS path name of the file, you can retrieve the file from the backup saveset.

Restoring several files

When you have to restore several files (such as entire directories or subdirectories) perform the following steps:

1. Restore only the container file from the backup saveset.

   If you do not remember the name of the container filesystem, look for a file with file type .CONTAINER as the extension.

2. Place the container in a directory with the same name. For example, if the container file is EARNINGS.CONTAINER, place it in [EARNINGS].

3. Bind the container to a UNIX pathname using UCX and export it. For example:

   ```
   UCX> BIND DUA0:[EARNINGS] "/earnings"
   UCX> ADD EXPORT "/earnings" /HOST=*
   ```

4. Mount the container file system on an NFS client. For example:

   ```
   mount vms:/earnings /mnt
   ```

5. Get a directory listing of files you want to restore from the file system you just mounted. You do this by setting the default directory on the NFS-client to the file system you just mounted and issuing a *ls -i* command, as follows:

   ```
   $ cd /mnt
   $ ls -i
   ```

   The *-i* option displays the inode numbers in the directory listing. You can also use the *cd* command to set the default directory to the subdirectories or use the *ls -R* to display recursive listing of files, as long as you do not access the files themselves.

   You could also write a UNIX shell script to generate a list of files with inode numbers (in hexadecimal), then trim it down to the specific files needed, then go back to the VMS system to restore the files.

4.8 Starting the NFS Server

The NFS server is started by the command procedure UCX$NFS_STARTUP.COM, which is invoked when the Connection is started by UCX$STARTUP.COM. These command procedures are located in the SYS$MANAGER directory.

The NFS server startup file defines a set of logical names that provide default characteristic values. You can permanently change these characteristics by editing the NFS startup procedure. You can use the SET NFS_SERVER command to change the parameters for the running server.

You can change the following parameters in the startup file:

- The logging of error messages

- The number of client hosts supported by the server
- The file inactivity timer
- The number of threads
- The size of cached transactions
- The value of the default UID
- The value of the default GID
- Security options
- Server time
- Server cache parameters
- NFS server process quotas
- Time differentials relative to GMT

Example 4–1 in Section 4.9 shows the UCX$NFS_STARTUP.COM file.

4.9 Tuning NFS Server Performance

The values of the logical names in the UCX$NFS_STARTUP.COM command procedure (shown in Example 4–1) affect the performance of the NFS server.

Example 4–1 UCX$NFS_STARTUP.COM File

```
$ !+
$ !             UCX$NFS_STARTUP.COM -- VMS/ULTRIX Connection Software,
$ !                                    NFS startup
$ !
$ !                          COPYRIGHT (C) 1988, 1989 BY
$ !                    DIGITAL EQUIPMENT CORPORATION, MAYNARD
$ !                     MASSACHUSETTS.  ALL RIGHTS RESERVED.
$ !
$ !  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
$ !  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
$ !  OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES
$ !  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER
$ !  PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
$ !
$ !  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND
$ !  SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
$ !
$ !  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
$ !  SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
$ !
$ !
```

(continued on next page)

Example 4-1 (Cont.) UCX$NFS_STARTUP.COM File

```
$ ! Modifications
$ !
$ ! mdk 11-Sep-89                NFS cannot start if the application license is
$ !                              not loaded, so don't attempt execution.
$ !
$ !                              Removed checks for shareable libraries.  This is
$ !                              done by UCX$UCP_STARTUP - so call it if license
$ !                              status is not known, or not using fast option
$ !
$ !
$ !                               Remove world from required_privileges
$ !
$ ! dsr 5-Apr-1990                Add "/EXE" to all "define"s
$ !
$ !
$ REQUIRED_PRIVILEGES = "DETACH,BYPASS,SYSPRV,SYSNAM,ALTPRI"
$ PREV_PRIVS = F$SETPRV(REQUIRED_PRIVILEGES)
$ IF .NOT. F$PRIVILEGE(REQUIRED_PRIVILEGES) THEN GOTO NO_PRIVILEGES
$ ON CONTROL_Y THEN GOTO EXIT
$ ON ERROR THEN GOTO EXIT
$ !
$ ! Re-start UCX if license status is not known, or called without fast option
$ !
$ UCX$START_UCX = 0
$ IF F$TYPE (UCX$LICENSE_STATUS) .EQS. "" THEN UCX$START_UCX = 1
$ IF P2 .NES. "FAST" THEN UCX$START_UCX = 1
$ IF .NOT. UCX$START_UCX THEN GOTO SKIP_UCP_STARTUP
$ @SYS$MANAGER:UCX$UCP_STARTUP
$ IF .NOT. $STATUS THEN GOTO EXIT
$SKIP_UCP_STARTUP:
$ IF .NOT. UCX$LICENSE_STATUS THEN GOTO NO_APPLICATIONS
$ !
$ ! Check to see if communication was started
$ !
$ IF F$GETDVI ("BG:","EXISTS") THEN -
     IF .NOT. F$GETDVI ("BG:","MNT") THEN GOTO NO_COMM
$ !
$ ! Initial Configuration for NFS Server
$ !
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS_TIME_DIFFERENTIAL 4   ❶
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_ERROR 1   ❷
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_HOSTS 20   ❸
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_UID -2   ❹
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_GID -2
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_INACTIVITY 02:00   ❺
```

(continued on next page)

Example 4–1 (Cont.) UCX$NFS_STARTUP.COM File

```
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_OPCOM 1          ❻
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_SECURITY 0       ❼
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_THREADS 20       ❽
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_XID 20           ❾
$ DEFINE/SYSTEM/EXE/NOLOG UCX$NFS00000000_DEBUG 0
$ !
$ EXPO_FILE = F$SEARCH ("SYS$SYSTEM:UCX$EXPORT.DAT")
$ PROX_FILE = F$SEARCH ("SYS$SYSTEM:UCX$PROXY.DAT")
$ !
$ DATA_FILE = F$TRNLNM ("UCX$EXPORT")
$ IF DATA_FILE .NES. "" THEN EXPO_FILE = F$SEARCH ("''DATA_FILE'")
$ IF EXPO_FILE .NES. "" THEN GOTO CHK_PROX
$ WRITE SYS$OUTPUT "WARNING - The NFS EXPORT file is missing."
$ !
$ CHK_PROX:
$ DATA_FILE = F$TRNLNM ("UCX$PROXY")
$ IF DATA_FILE .NES. "" THEN PROX_FILE = F$SEARCH ("''DATA_FILE'")
$ IF PROX_FILE .NES. "" THEN GOTO END_CHECK
$ WRITE SYS$OUTPUT "WARNING - The NFS PROXY file is missing."
$ !
$ END_CHECK:
$ !
$ ! Define UCX filesystems
$ !
$ SET NOON
$ T = F$SEARCH ("SYS$MANAGER:UCX$NFS_SET_FS.COM")
$ IF T .EQS. "" THEN -
 WRITE SYS$OUTPUT "Warning - No filesystems defined for NFS Server"
$ IF T .NES. "" THEN -
        @SYS$MANAGER:UCX$NFS_SET_FS.COM
$ !
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_CACHE_SIZE  256        ! 256x8KB total  ❿
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_CACHE_LOW_LIMIT  4     ! 4 buffers min free
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_CACHE_HIGH_LIMIT 8     ! 8 buffers goal
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_TRANSFERSIZE 8192
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_WRITESIZE 512
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_SHOW_VERSION 0  ⓫
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_MODUS_OPERANDI 64      ! Disable lock twice msgs
$ DEFINE/SYSTEM/EXE/NOLOG UCX$CFS_FATAL_MESSAGES "_OPA0:"
$                                                        ! Fatal msgs on console
$ !
$
$ IF P1 .EQS. "" THEN P1 = "SYS$SPECIFIC:<SYSMGR>UCX$NFS_LOGFILE.LOG"  ⓬
$ !
$ ! Define/sys ucx$nfs_host ucx$inet_host or ucx$inet_cluster
$ !
$ HOST = F$TRNLNM("UCX$INET_HOST")
$ IF HOST .EQS. "" THEN GOTO NO_COMM
$ !
```

(continued on next page)

Example 4–1 (Cont.) UCX$NFS_STARTUP.COM File

```
$ ON ERROR THEN GOTO NO_RUN
$ IF F$SEARCH ("''P1'") .NES. "" THEN -
  PURGE/NOLOG/KEEP:4 'P1'
$ RUN SYS$SYSTEM:UCX$SERVER_NFS.EXE/DETACH -
        /OUTPUT=NLA0: -
        /ERROR='P1' -
        /AST_LIMIT=512 -
        /BUFFER_LIMIT=100000 -
        /EXTENT=10000 -
        /FILE_LIMIT=1024 -  ⓭
        /IO_BUFFERED=400 -
        /IO_DIRECT=200 -
        /QUEUE_LIMIT=64 -
        /ENQUEUE_LIMIT=1500 -
        /MAXIMUM_WORKING_SET=10000 -  ⓮
        /PAGE_FILE=10000 -
        /PRIORITY=8 -
        /PRIVILEGES=(BYPASS,SYSPRV) -
        /UIC=[1,3] -
        /NORESOURCE
$EXIT:
$ PREV_PRIVS = F$SETPRV(PREV_PRIVS)
$ EXIT
$NO_PRIVILEGES:
$ WRITE SYS$OUTPUT "Insufficient privileges to start the NFS Server"
$ WRITE SYS$OUTPUT "NFS Server Requires ''REQUIRED_PRIVILEGES' privileges."
$ GOTO EXIT
$NO_SHARED:
$ WRITE SYS$OUTPUT "Failed to install required shareable images"
$ WRITE SYS$OUTPUT "NFS Server requires ''LIBRARY'"
$ GOTO EXIT
$NO_COMM:
$ WRITE SYS$OUTPUT "Internet Network not active"
$ WRITE SYS$OUTPUT "NFS Server not started"
$ GOTO EXIT
$NO_RUN:
$ WRITE SYS$OUTPUT "Failed to start NFS Server image"
$ GOTO EXIT
$NO_APPLICATIONS:
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "The UCX license is not active on this system"
$ WRITE SYS$OUTPUT "The NFS Server will not be started"
$ GOTO EXIT
```

The following descriptions refer to Example 4–1.

❶ Time differential

The logical name UCX$NFS_TIME_DIFFERENTIAL is given a value in hours that represents the offset from Greenwich mean time. A positive number means your time zone is west of Greenwich (for example, North and South America) and a negative number means it is east of Greenwich (for example, Europe).

❷ Error messages

The NFS server supports the logging of error messages into a log file. The name of the log file is specified by the NFS startup command file. When the NFS server starts, it always creates a new log file.

If you use EDT to display error messages, you must disable logging to the log file by issuing the SET NFS_SERVER/DISABLE=ERROR command, which flushes the buffer and closes the file. You can also use the VAX Language-Sensitive Editor (LSE) to look at the error log. If you use LSE without disabling the log file, you will not see the latest error messages logged to the file.

When logging is reenabled with the SET NFS_SERVER command, the NFS server checks for the existence of the log file and then does one of the following:

- If the log file does not exist, it creates a new file.
- If the log file exists, the NFS server continues to write into existing file.

While logging is disabled, all error messages are lost.

Table 4–2 explains the function of the logical names that control the error messages.

Table 4–2 Logging Error Messages

Logical Name	Description
UCX$NFS00000000_ERROR	Defines whether error messages output is enabled or disabled.
UCX$NFS00000000_OPCOM	Defines whether messages are sent to the operator console.

You can enable or disable error messages temporarily with the SET NFS_SERVER command. See Chapter 5 for a description of the SET NFS_SERVER command.

❸ Maximum number of client hosts

The UCX$NFS00000000_HOSTS parameter defines the maximum number of client hosts that can be defined within the NFS server. During NFS initialization, the server builds a host table from the proxy database file. If the number of client hosts listed in the proxy database exceeds this number, the excess client host names are ignored. This action has the effect of disabling access to the server for those client host names that cannot be cached.

You set this parameter to control the size of the host table within the NFS server. This parameter should be large enough to allow for the definition of all the hosts present in the proxy database.

Making the parameter value larger than what is needed makes the NFS server allocate redundant virtual memory within the NFS server.

Because this parameter value is static, the NFS server has to be restarted for parameter value changes to take effect.

❹ UID and GID values

These values are mapped to the NFS client when the client accesses the NFS server with *root* access. These values are used only when the client accesses the ULTRIX file system to determine whether or not ULTRIX file protection checking is done. If the values are set to UID=0 and GID=1 (which indicates *root* in ULTRIX), then no ULTRIX file protection checking is done. If the values are set to any other values, ULTRIX file protection checking is done. By default they are set to –2.

Regardless of the UID and GID values that are set, file access is determined to a large extent by the privileges assigned to the VMS account that the NFS client is mapped to in the proxy database.

You can set these default parameters dynamically with the SET NFS_SERVER /UID_DEFAULT=0 /GID_DEFAULT=0 command.

❺ Inactivity timer interval

Because the NFS server is stateless, the server does not maintain state about any of its clients. Consequently, clients do not send explicit open and close file requests to the server; instead, the VMS operating system opens and closes the files for the clients. The server caches the open files to create internal state for each file within the NFS server environment.

Files must be closed to allow VMS users access to the files through RMS. The server uses the following guidelines to close the files:

- Close the files that were not accessed during the specified time interval.
- Close the oldest files when a user wants to access a new file and the maximum number of opened files for the NFS process or the maximum number of opened files specified in the VMS account is reached.

The NFS server keeps an activity timestamp for each opened file to help manage the open file cache. This time interval is specified to the NFS server by means of the logical name UCX$NFS00000000_INACTIVITY. This logical name is loaded at NFS server startup; however, you can modify it dynamically with the UCX command SET NFS_SERVER. To do this you use the /INACTIVITY_TIMER qualifier, which enables you to specify the time interval in minutes and seconds.

The default setting for this value is 02:00, or two minutes. Making the interval too short causes the NFS server to close files more often, thereby, reducing performance.

❻ OPCOM

This logical name enables messages to be sent to the operators console. It is enabled (set to 1) by default. To disable, set this value to 0.

❼ Security

The logical name UCX$NFS00000000_SECURITY represents a bit-mask value, where each set bit adds a new security feature to the NFS operation. The meaning of the security mask bits are listed in Table 4–3. See Section 4.6 for more information.

Table 4–3 Security Logical Name Mask Bits

Bit	Set Bit Description
1	Enables security (UCX$NFS_REMOTE identifier).
2	Only superusers can remotely mount the file system on the NFS server.
3	Only privileged ports on the client host can send messages to the NFS server.
4	Enables you to use information in the SYSUAF file to restrict network access.

❽ Number of threads

The NFS server is an asynchronous, multithreaded process, which means that multiple NFS requests can be processed concurrently. Each NFS request is referred to as a **thread**, and exists from the reception of the NFS request from the network until the corresponding response is sent back to the client. The UCX$NFS00000000_THREADS value determines the maximum number of threads that can be simultaneously active.

The performance of the NFS server is directly related to this value. Assuming the host server system has the available resources (CPU, memory, and disk speed), you can raise this value to allow for increased NFS activity.

If your clients still experience NFS server timeout conditions after you have increased this value, you may want to consider using a higher-performing server. Alternately, you can increase the NFS timeout period on the clients by using the time option of the *mount* command

The recommended value for an average load is a thread maximum of 20. This value is static, which means that the NFS server must be restarted for this parameter value to take effect. Also, if you increase this parameter, you may need to increase the /PAGE_FILE parameter in the NFS startup file as well.

The SHOW NFS_SERVER command displays the maximum number of threads that have ever been active since the server has been activated. Use this display to tune this parameter. If you increase this value, it is recommended that you also increase the XID cache value.

❾ XID cache

The NFS service uses UDP/IP, which does not guarantee the delivery of the NFS request between the client and the server. Because of the nature of the protocol, the client periodically reissues the NFS request if a response has not been received within a certain time period. Each request is assigned an XID value, which identifies the transmission request. Whenever a request with the same XID value is received by the NFS server, it is recognized as a duplicate XID. The NFS server ignores the retransmission because the response to that particular message is currently being processed by the NFS server.

If the XID value is not a duplicate value, the message is dispatched and processed, because the preceding messages transmitted by the client have been lost.

If the response of the NFS server has not been received by the client, a duplicate XID value is not detected because the response has not been sent to the client. Consequently, the CREATE, DELETE, LINK, RENAME, SYMLINK, and SETATTR operations will successfully complete the first time but fail the second time with a false error. To prevent this false error, the NFS server maintains a cache of these requests, which are identified by their hostname (client system) and XID value. When a request for one of these operations is received, the queue is checked for a first attempt response. If it is a first time, the response is sent to the client; otherwise, a normal operation is started.

The size of the cache for these operations is limited by the value of the logical name UCX$NFS00000000_XID. Depending on the frequency of these operations, the size of the cache is critical. The recommended value

is 20, but it is recommended that you increase this value if you notice any of the following situations on a client:

- A file or link creation request returns "file exists," but the file did not exist when the file creation was requested and does exist after the file creation request.
- A file/directory or link deletion request returns "no such file," but the file did exist, and does not exist after the deletion request.
- A rename request returns "no such file," but the file did exist under its old name when the rename operation was requested and exists under its new name after the rename request.
- A set attributes request returns "privilege violation," but the file was writable by either the owner or group requestor, and does exist after the request with the new attributes.
- You increase the maximum number of threads.

The XID value is static, which means that the NFS server must be restarted for an XID parameter changed to take effect. Also, if you increase this parameter, you may need to increase the /PAGE_FILE parameter in the NFS startup file as well.

❿ This is the upper limit on the amount of memory consumed by a data cache. You might want to make this 10 percent of physical memory. If you change this parameter, you also need to change the page file and maximum working set parameters on the RUN command.

⓫ Version numbers

The UCX$CFS_SHOW_VERSION logical name controls whether version numbers are displayed with the file names, when there is only one version of the file. Normally, the VMS NFS server removes the version number if there is only one version of the file. To have the VMS NFS server always display a version number, regardless of the number of files, set the logical to 1. The default (0), specifies that the NFS server will not display a version number if there is only one version of the file. If there is more than one version of the file, the version numbers are always displayed.

⓬ P1

The P1 parameter is used as the name for the NFS error logging file.

⓭ File limit — NFS process channel limit

Because the NFS service is a VMS process, it is limited by certain quotas. One such quota is FILE_LIMIT. Make this quota value high enough to allow the NFS server to keep the files of all clients open. Whenever the server is forced to close a file because this limit has been reached, it outputs a message to its log file.

FILE_LIMIT is determined by the system parameter CHANNELCNT. No matter how large the value of FILE_LIMIT, the VMS operating system uses the value of the CHANNELCNT parameter as an absolute maximum. Therefore, it may be necessary to adjust the CHANNELCNT parameter and reboot the VMS system to increase the maximum number of channels.

⓮ NFS process memory limit

The RUN command includes parameters to control the size of virtual memory and pagefile allocation for the (detached) NFS server process. If insufficient memory is available, the server may be forced to deaccess files in favor of new files. Insufficient memory or excessive paging may result in decreased server performance.

To improve server performance, set the memory limits (/WORKING_SET, /EXTENT, and /PAGE_FILE) appropriately. If you increase the maximum number of threads or if you increase the XID cache, you may need to increase these values. For these values to take effect, you must restart the NFS server. You may need to increase the SYSGEN parameter WSMAX to accommodate certain values for the above parameters. For the WSMAX parameter to take effect, you must reboot the system.

4.9.1 UAF File Limits

VMS user accounts specify the maximum number of files that can be accessed simultaneously with the FILLM parameter. When an NFS request is made, the client's identity is mapped to a VMS account by means of the proxy database. The specified VMS user name is used to obtain authorization information from the system. If a file must be accessed to process a read or write request, the file access cannot exceed the FILLM quota as specified in the VMS user's authorization information.

To make this application of file limit transparent to the client, the oldest (least accessed) file is closed, and a new file is accessed. An indication of when this happens can be seen in the error log file, which will contain an exceeded quota message.

If the FILLM quota is set too low, it will degrade the NFS server's performance.

Under normal situations, the NFS request is successfully completed once the quota limit is corrected. The INACTIVITY timer interval can reduce the required open files quota by closing old files automatically before the contention.

Because the open files quota information is loaded into the NFS server at startup time, any changes to a user's authorization information will not take effect until the NFS server is restarted.

Each NFS client is defined by a unique UID and host name combination. This information is stored within the proxy database. The database provides for translation from UNIX UID/host to VMS user name. It is possible for multiple UID/host entries to translate into a single VMS user name. Because the file limits are managed within the framework of the VMS user name, multiple clients can share the same file quota. Therefore, ensure that the FILLM value for the VMS user name is sufficient to provide ample performance for the client associated with that user name.

Additionally, make certain that the CHANNELCNT parameter (which sets the maximum number of channels that can be used by a process) is large enough to handle the total number of files accessed by all the NFS clients connected to the server.

4.9.2 Displaying NFS Server Tuning Information

You can use the SHOW NFS_SERVER command to display information about the NFS server that you can use in tuning the performance of the server. The counter information is especially useful.

You can issue the SHOW NFS_SERVER command to display information for a specific client or host. If the command issued for a host, the host must be specified in the proxy database for the command to work correctly.

Example 4–2 shows a SHOW NFS_SERVER display. The numbers in the example are keyed to the discussion that follows.

Example 4–2 SHOW NFS_SERVER Command Display

```
Server:                                   Loaded: 14-JUN-1989 15:35:01.73

Status:     ACTIVE                        Running:          0 00:24:21.26

Memory allocated ❶          232436          RPC errors ❷
Message processing:                   Authentication                    0
  Threads busy ❸                0         Others                        0
  Threads free                 15   Mount data base:
  Max. threads busy             5     Mounted file systems ❹            1
  Duplicate cache xid ❺ 0            Current users                     1
  Duplicate active xid          0     Maximum mounted                   1
  Dropped                       0     Maximum users                     1
Data exchange:                        NFS operations: ❻
  Bytes sent ❼            11839124     null              0  getattr     42
  Bytes rcvd              10900824     setattr          12  lookup     186
  Messages sent               2956     readlink          0  rename       0
  Messages rcvd               2956     read           1417  write     1284
  Max. message sent           8292     statfs            1  create       2
  Max. message rcvd           8328     remove            1  link         0
Open files:                            symlink           0  mkdir        1
  Maximum opened ❽               2     rmdir             1  readdir      7
  Closed per interval            0   Total NFS operations             2954
  Currently opened               0   Error messages ❾                    0
```

❶ The memory allocation shows the amount of memory that the NFS server is using at a given instant. The amount of memory allocation is affected by the number of threads, size of the volatile user database, mounted file system structures, and number of opened files.

❷ The RPC errors counter shows the number of errors for the following: authentication (potential intruder attempts), garbled RPC messages, wrong RPC protocol number, no procedure, no program, and version mismatch. The error logging file contains the information about the hosts and users that caused the erroneous RPC message.

❸ The multithreading counters show the number of threads being processed at any given instant. The number of threads includes the threads queued to the Internet driver for receiving messages and the maximum number of threads that have been simultaneously active in process. Whenever the value of the multithreading counters equals the specified value of the threads in the NFS startup file, it indicates that load on the server is too high for the specified maximum number of threads. You may want to increase the maximum number of threads to improve NFS performance.

❹ The mount database counters display the number of mounted file systems and clients currently active within the NFS server. Because of the stateless nature of the NFS server, these counters may contain an inaccurate value.

❺ The duplicate/dropped message counter displays the number of duplicate XIDs and dropped messages. The XID amount includes the duplicate XIDs present during message processing and caching, and the number of messages dropped because of busy resources within the VMS environment. A large value indicates that the NFS performance is not good enough to avoid client timeout and message retransmissions. You may want to increase the number of threads to improve NFS performance.

❻ The operation statistics counter displays the number of NFS operations for each operation.

❼ The data exchange counter shows the amount of bytes transmitted and received by the NFS server. This value provides an indication of the amount of load on the NFS server.

❽ The open file counters display the maximum number of simultaneously open files, currently open files, and close operations. A high number of files closed per activity time interval means that the clients have simultaneously accessed a large number of files. Accessing a large number of files may affect NFS server performance because of many file-close operations. You may want to increase the inactivity timer value to improve NFS performance.

The value of the maximum simultaneously opened files indicates what you should use for an open channel quota value.

❾ The error message counter displays the number of error messages in the error log. To examine the error log, you must first disable logging with the following command:

```
UCX> SET NFS_SERVER /DISABLE=ERROR
```

After examining the error file, reenable logging with the following command:

```
UCX> SET NFS_SERVER /ENABLE=ERROR
```

4.9.2.1 Monitoring System Services The SHOW CFS command is useful for monitoring the distribution of the Connection file system services and the consumption of system time by the various system services. See Chapter 5 for a detailed description of the SHOW CFS command. Example 4–3 is an example of a SHOW CFS display. The numbers in the example are keyed to the discussion that follows.

Example 4–3 SHOW CFS Command Display

```
                                CFS SERVICES                31-AUG-1989 14:10:02.74
  CFS Services❶                     VAX/VMS System Services❷
-----------------------         ---------------------------------------------------
CLOSE                 0
CREATE_FH             0         $ASSIGN              0   $QIO                      0
CREDIR_FH             0         $DASSGN              0        Access               0
FREEBUFF              0                                       Create               0
GETATTR               0         $DEQ                 0        Deaccess             0
LINK_FH               0         $ENQ                 1
LOOKUP_FH             0                                       Read_attr            0
OPEN_FH               0         $EXPREG              2        Write_attr           0
READ                  0         $SETPRT              0
READBUFF              0                                       Lookup               0
READDIR_FH            0
$CLREF                0
READLINK_FH           0         $SETEF               0        Extend               0
REMDIR_FH             0
REMOVE_FH             0         $DCLAST              0        Delete               0
RENAME_FH             0         $CLRAST              0        Enter                0
SETATTR               0         $SETAST              7        Remove               0
STATFS                0
SYMLINK_FH            0         $GETDVI              0        Read_V               0
WRITE                 0                                       Write_V              0
OTHER                 4         $CHKPRO              0
TOTAL                 4
```

❶ The summary of services provides an indication of the distribution of the CFS services. The OTHER category includes BIND, EXPORT, IMPORT, OPERATOR, STATUS, TRANSLATE, and UNBIND services.

❷ The summary of VMS system services provides an indication of the consumption of various system services. Each service used by CFS is tallied with the addition of the subtotals for the $QIO system service. These totals are indicators of the XQP and disk performance.

4.9.2.2 Monitoring Connection File System Performance The SHOW CFS /SUMMARY command provides a good indication of Connection file system performance at the moment, by providing the current and maximum values. See Chapter 5 for a description of the SHOW CFS/SUMMARY command. Example 4–4 is an example of a SHOW CFS/SUMMARY display. The numbers in the example are keyed to the discussion that follows.

Example 4–4 SHOW CFS/SUMMARY Command Display

```
                    CFS Service status and performance    31-AUG-1989 14:10:18.14
❶                                   ❷
Service State    Cur   Max    Total   Cacheop   Cur   Hit   I/O   Inc     Status
--------------   ---   ---   ------   -------   ---   ---   ---   ---   ------------
FP-Access          0     0        0   Read        0     0     0     0   Clusize    16
FP-Attributes      0     0        0   Read-A      0     0     0     0   Limit     256
FP-Delete          0     0        0   Write       0     0     0     0   Inuse       0
FP-Dir             0     0        0   Write-A     0     0     0     0   Busy        0
FP-Rename          0     0        0   Write-D     0                     Hitrate     0
Broadcast          0     0        0   ❸
Buffered           0     0        0   Nameop    Cur   Hit     Status
Internal           0     0        0   ------    ---   ---   ------------
I/O                0     0        0   Add         0       Tabsize     0
Lock               0     0        0   Delete      0
Inuse              0                  ❹
Logging            0     0        0   Lookup      0     0   Hitrate     0
Resource           0     0        0
RMS                0     0        0   Fileop    Cur   Hit     Status
Service            0     1        1   ------    ---   ---   ------------
Timer              0     0        0   Find        0     0   Limit       0
Other              0     0        0   Find-A            0   Inuse       0
Synch              0     0        0   Find-C            0   Timeout     0
                                                           Hitrate     0
❺                                                                      ❻
Services count     1     1        5   ATCBs:   0 TBABs:   0 RDCBs:   0  Pages:   0
```

❶ The summary of wait states indicates the reason why a thread goes into a wait state. By determining where most of the waits occur, you can determine what in the system is slowing down the server.

❷ The summary of data cache performance indicates how the CFS cache is performing. This summary provides the current set of parameters: cluster size, the cache size limit (UCX$CFS_CACHE_SIZE[1]), how much of the cache is in use, how many I/O requests are outstanding, and the overall hit rate of the cache. All these values are related to cluster size (clusize) buffers. The cluster size buffers show the maximum number of blocks that can be read from or written to a disk.

The read and write groups are synchronous, while read-a and write-a groups are asynchronous. The write-d is a deferred write operation, where the caller issues an unmap of data without writing to disk. UFS uses the write-d frequently.

❸ The summary of name cache performance provides the number of lookup operations that were saved by the name cache. The name cache converts a file-handle/name pair into another file-handle that saves lookup operations because each entry found in the cache (a hit) saves a lookup operation.

[1] This logical name is defined in the NFS server startup file, UCX$NFS_STARTUP.COM.

❹ The summary of file-attributes cache indicates how the attributes (file RDCB) cache is performing. The file-attributes cache values are described in Table 4–4.

Table 4–4 File-Attributes Cache Values

Value	Description
Limit	Maximum number of RDCBs that can be in cached status at any moment
Inuse	Number of RDCBs in use
Timeout	Number of seconds that any cached file RDCB can exist
Hit rate	Number of XQP QIOs (read attributes) that were saved because the valid RDCB was found (other overhead is also saved, for example, $ENQ/$DEQ time)

❺ The summary of services and resources allocated provides the number of services and varying control blocks that are allocated. These services may not correspond one-for-one to NFS services because one service can call another to perform a special function. This is particularly true for UFS file systems, which make internal calls to perform certain operations in the VMS space.

The services and resources allocated values are described in Table 4–5.

Table 4–5 Services and Resources Allocation Values

Value	Description
ATCBs	Asynchronous thread control blocks (one per thread)
TBABS	Thread backout attachment blocks (one per active virtual lock)
RDCBs	Resource description and control blocks (one per node, one per file system, one per file, one per cache buffer)
Hit rate	Number of XQP QIOs (read attributes) that were saved because the valid RDCB was found (other overhead is also saved, for example, $ENQ/$DEQ time)

❻ Pages are random pages from the virtual address space that are used dynamically when 512 bytes or less are needed for long-lasting scratch space. A page is currently allocated onto the file RDCB to store any ACL information. It also provides device ACL support by attaching a page onto the file-system RDCB.

4.10 Stopping the NFS Server

The NFS server is stopped by the procedure UCX$NFS_SHUTDOWN.COM (located in SYS$MANAGER), which is automatically invoked if you shut down the Connection using UCX$SHUTDOWN.COM.

You can stop the NFS server even though clients still have file systems mounted on the server. If a client has a file system mounted with the option *hard*, and the client accesses the file system while the server is down, the client will stall while it is waiting for a response from the server.

Alternatively, if the client has a file system mounted *soft*, the client will receive an error message if it attempts to access a file system while the server is down.

Because the NFS protocol is stateless, clients with file systems mounted on the server do not need to remount when the server is restarted. To ensure this uninterrupted service, you must be sure to bind all your Connection file systems prior to restarting the NFS service. Example 4–5 shows the UCX$NFS_SHUTDOWN.COM file.

Example 4–5 UCX$NFS_SHUTDOWN.COM File

```
$ !+
$ !   UCX$NFS_SHUTDOWN.COM -- VMS/ULTRIX Connection Software, NFS shutdown
$ !
$ !                          COPYRIGHT (C) 1988 BY
$ !                     DIGITAL EQUIPMENT CORPORATION, MAYNARD
$ !                      MASSACHUSETTS.  ALL RIGHTS RESERVED.
$ !
$ !  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
$ !  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
$ !  OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES
$ !  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER
$ !  PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
$ !
$ !  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND
$ !  SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
$ !
$ !  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
$ !  SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
$ !
$ REQUIRED_PRIVILEGES = "WORLD,SYSPRV"
$ PREV_PRIVS = F$SETPRV(REQUIRED_PRIVILEGES)
$ IF .NOT. F$PRIVILEGE(REQUIRED_PRIVILEGES) THEN GOTO NO_PRIVILEGES
$ ON CONTROL_Y THEN GOTO EXIT
$
$ ON ERROR THEN GOTO EXIT
$ PROCNAM = P1
$ IF PROCNAM .EQS. "" THEN PROCNAM := NFS$SERVER
```

(continued on next page)

Example 4–5 (Cont.) UCX$NFS_SHUTDOWN.COM File

```
$ CONTEXT = ""
$ LOOP:
$   P1 = F$PID(CONTEXT)
$   IF P1 .EQS. "" THEN GOTO NO_NFS
$   IF F$GETJPI(P1,"PRCNAM") .NES. PROCNAM -
    THEN GOTO LOOP
$
$ STOP/ID= 'P1'
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_ERROR/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_GID/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_HOSTS/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_INACTIVITY/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_OPCOM/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_OUTPUT/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_SECURITY/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_THREADS/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_UID/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_XID/user
$ DEASS/SYSTEM/EXE UCX$NFS"'p1'"_DEBUG/user
$ WRITE SYS$OUTPUT "''PROCNAM' stopped"
$EXIT:
$ PREV_PRIVS = F$SETPRV(PREV_PRIVS)
$ EXIT
$NO_PRIVILEGES:
$ WRITE SYS$OUTPUT "Insufficient privileges to STOP the NFS Server"
$ WRITE SYS$OUTPUT "NFS Server Requires ''REQUIRED_PRIVILEGES' privileges."
$ GOTO EXIT
$ NO_NFS:
$ WRITE SYS$OUTPUT "''PROCNAM' not found"
$ GOTO EXIT
```

4.11 Troubleshooting the NFS Server

When an NFS client experiences a problem using the NFS server on the VMS operating system, try to isolate where the problem is occurring. You may first want to check the following:

- Establish that the two hosts are able to communicate. On a VMS system, you can use the LOOP command to test the connectivity between the two systems.
- If the LOOP command fails, recheck your physical network connections and your Internet parameters.
- Make sure the Internet network and broadcast masks are correct for your interface.
- Check that the host addresses are correct in the HOST database.

However, if the two systems are able to communicate and the problem persists, determine whether the problem is being detected on the server or on the client. If the problem is occurring on the server, the NFS error count field in the SHOW NFS_SERVER display is incremented. The NFS server logs extended error messages to the NFS log file, UCX$NFS_LOGFILE.LOG (located in SYS$MANAGER), if logging has been enabled.

To examine this file, you must first disable logging. For example:

```
UCX> SET NFS_SERVER /DISABLE=ERROR
UCX> EXIT
$ TYPE SYS$MANAGER:UCX$NFS_LOGFILE.LOG
```

See Appendix A for a description of the NFS server error messages and suggested actions.

When you determine the cause of the error, reenable logging with this command:

```
UCX> SET NFS_SERVER /ENABLE=ERROR
```

The following errors are commonly encountered on the server:

- Modifying the NFS proxy database but not restarting the server.
- Creating an NFS proxy account with either a nonexistent or disabled VMS account.
- Forgetting to create proxy accounts for clients UID=–2, GID=–2 and UID=0, GID=1.
- Creating a HOST database entry in uppercase rather than lowercase.
- Not binding the file system before the NFS mount request.
- Not binding the underlying VMS volume on which the ULTRIX file system resides.

Whenever you modify the NFS server parameters using the UCX commands, messages are logged to the log file.

Also, examine the VMS operator log file for any Connection-related messages. Certain fatal messages are logged only to this file, prior to the exit of the NFS server.

The NFS server provides two checks; one on the server and one on the client. If no error is recorded on the server, then the problem is occurring on the client. The most common error occurring on a client is a permission-denied error. If this error occurs and the server has not recorded a privilege violation, one possiblity is that the VMS server returned the file's attributes as UID=–2, GID=–2, which means that the client does not have the proper access to the file. This problem occurs when the VMS server is unable to translate a VMS

file's attributes to a UID/GID pair from the proxy database. You can correct this problem by adding a proxy record to the database for the target VMS UIC.

Finally, the NFS protocol defines the set of error messages that can be returned by the server. The NFS server maps VMS system errors into NFS error messages. The NFS error messages are the messages that NFS clients see. Table 4–6 lists the NFS errors. There is not a one-to-one mapping between the VMS system error messages and the NFS error messages. For example, both the SS$_EXDISKQUOTA and SS$_EXFILLM messages map into the NFS_DQUOT error message. Any VMS system error message that is not listed in Table 4–6 is mapped into the NFS_IO message.

Refer to the VMS general-user documentation for a description of the VMS error messages.

Table 4–6 Mapping VMS System Error Messages to NFS Error Messages

VMS System Error	NFS Error
SS$_BADFILENAME	NAMETOOLONG
SS$_BADIRECTORY	NOTDIR
SS$_DEVICEFULL	NOSPC
SS$_DEVOFFLINE	NODEV
SS$_DIRNOTEMPTY	NOTEMPTY
SS$_DUPFILENAME	EXIST
SS$_ENDOFVOLUME	NOSPC
SS$_EXDISKQUOTA	DQUOT
SS$_EXFILLM	DQUOT
SS$_EXQUOTA	ACCES
SS$_NOIOCHAN	DQUOT
SS$_NOPRIV	ACCES
SS$_NORMAL	NFS_OK
SS$_NOSUCHDEV	NODEV
SS$_NOSUCHFILE	NOENT
SS$_TOOMUCHDATA	FBIG
Any other VMS error	IO

4.12 Running the NFS Server on a VMS Internet Cluster

The NFS server application automatically responds to the requests it receives on any active Internet interface. Therefore, if several of your VAXcluster hosts have Internet cluster interfaces, the NFS server can execute as a clusterwide application. Clients who mount file systems using the cluster Internet host name can then be served by any of the NFS servers in the cluster. Also, if one of the servers is taken down, client requests are redirected to another host in the cluster.

5

VMS/ULTRIX Connection Commands

This chapter describes all of the VMS/ULTRIX Connection commands. The following sections list the Connection commands by task to help you identify which commands are related to the tasks you want to complete. If you need more information on a command, see the individual description in the alphabetized list in Section 5.6 .

5.1 Connection File System Commands

The Connection file system (CFS) commands are used to create, delete, and maintain Connection file systems. There are two types of Connection file systems: ULTRIX file systems and VMS file systems. A VMS file system is a collection of VMS files and directories on a mounted VMS volume. An ULTRIX file system is a collection of ULTRIX files and directories hosted on a VMS volume.

The file structures for a single ULTRIX file system are kept in a single VMS data file, called a container file. Each ULTRIX user data file is stored as a VMS data file using a CFS-encoded VMS file name.

The Connection file system commands are:

ANALYZE CONTAINER	BIND
CREATE CONTAINER	CREATE DIRECTORY
DELETE CONTAINER	DIRECTORY
EXPORT	IMPORT
REMOVE DIRECTORY	REMOVE FILE
SHOW BIND	SHOW CFS
UNBIND	

5.2 Internet Commands

The Internet commands are used to configure and control the Internet network, show the status of the network, and test network components.

Some of these commands refer to different objects of the network: local host, remote host, network interface, and the network as seen by the Connection software (commands that treat the network as one virtual host).

The Internet commands are:

LOOP	SET ARP
SET COMMUNICATION	SET HOST
SET INTERFACE	SET NAME_SERVICE
SET NETWORK	SET ROUTE
SHOW ARP	SHOW COMMUNICATION
SHOW DEVICE_SOCKET	SHOW HOST
SHOW INTERFACE	SHOW NETWORK
SHOW PROTOCOL	SHOW ROUTE
START COMMUNICATION	START ROUTING
START SERVICE	STOP COMMUNICATION
STOP ROUTING	STOP SERVICE
ZERO INTERFACE	ZERO PROTOCOL

5.3 NFS Commands

The NFS commands are used to control and monitor NFS server performance.

The NFS server acts on file requests made by Connection clients. It supports the NFS, MOUNT, and PORTMAPPER service functions.

The NFS commands are:

ADD EXPORT	ADD PROXY
REMOVE EXPORT	REMOVE PROXY
SET NFS_SERVER	SHOW EXPORT
SHOW NFS_SERVER	SHOW PROXY
ZERO NFS_SERVER	

5.4 Database Commands

The database commands can be used for the following functions:

- Converting ULTRIX databases into VMS databases
- Converting VMS databases into ULTRIX databases
- Creating new and empty databases

By default the Connection databases are located in SYS$COMMON:[SYSEXE]. Table 5–1 provides more information on the Connection databases.

Table 5–1 Connection Databases

Default Name	Default Logical Name	Function
UCX$EXPORT.DAT	UCX$EXPORT	Specifies which file systems can be mounted from remote hosts.
UCX$HOST.DAT	UCX$HOST	Associates host names with Internet addresses.
UCX$NETWORK.DAT	UCX$NETWORK	Associates network names with network numbers.
UCX$PROXY.DAT	UCX$PROXY	Provides VMS identities for NFS clients.
UCX$ROUTE.DAT	UCX$ROUTE	Defines Internet gateways.

If you define a system-wide logical name for a database, the UCX database commands use the logical name during operations. If a system-wide logical name is not defined, the commands search your current directory for the database.

The database commands are:

CONVERT/VMS HOST	CONVERT/VMS NETWORK
CONVERT/VMS PROXY	CONVERT/ULTRIX HOST

CONVERT/ULTRIX NETWORK	CREATE EXPORT
CREATE HOST	CREATE NETWORK
CREATE PROXY	CREATE ROUTE

5.5 Miscellaneous Commands

The Connection also provides the following miscellaneous commands:

- EXIT
- HELP

5.6 UCX Command Reference

This section provides reference information on the UCX commands.

ADD EXPORT

Specifies the list of clients that can access the exported file system.

Format

ADD EXPORT "/ultrix_directory_spec"

Command Qualifier	Default
/HOST=(host_name[,...])	None.

Restrictions

You cannot use wildcards within the ULTRIX directory specification.

This operation requires read/write access to the UCX$EXPORT database.

Parameter

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name, and zero or more directory names separated by slashes.

Description

Only directories within file systems that have been exported on the server are eligible for remote mount from client workstations. Using the ADD EXPORT command, you specify the list of clients that can mount the exported file system.

The underlying file system must be bound if the client is going to mount it.

Command Qualifier

/HOST=(host_name[,...])
Specifies which remote hosts are allowed to access a file system.

If you specify /HOST=*, the file system is accessible from all hosts.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

Examples

1
```
UCX> ADD EXPORT "/golden/gate" /HOST=GOLDEN
```

Allows an ULTRIX client on host GOLDEN to access the Connection file system "/golden/gate".

2
```
UCX> ADD EXPORT "/golden/gate" /HOST=(GOLDEN,BRKLYN)
```

Allows an ULTRIX client on hosts GOLDEN and BRKLYN to access the Connection file system "/golden/gate".

3
```
UCX> ADD EXPORT "/golden/gate" /HOST=*
```

Allows an ULTRIX client from any host to access the Connection file system "/golden/gate".

ADD PROXY

Provides an account (VMS identity) for each NFS client.

Format

ADD PROXY vms_account

Command Qualifiers	Defaults
/GID=gid	None.
/HOST=(host_name[,...])	None.
/PERMANENT	See text.
/UID=uid	None.

Restrictions

This operation requires read/write access to the UCX$PROXY database.

Parameter

vms_account
Specifies the VMS account on a VMS system.

Description

ULTRIX and VMS use different authorization methods. It is not possible to provide VMS privileges to an ULTRIX client based on that client's identification. The ADD PROXY command provides a VMS identity (account) for each NFS client. This VMS identity is used as the basis for access control whenever the NFS client is accessing a VMS file.

Each ULTRIX client does not have to have a unique VMS account. It may be desirable to have a group of ULTRIX clients use the same VMS account.

The *vms_account* parameter is an account name found in the VMS UAF file.

Command Qualifiers

/GID=gid
Specifies the ULTRIX GID for the client that is to be given access.

/PERMANENT
Specifies that the proxy record is added to the permanent, on-disk database only.

If NFS is not active, the changes are made to the permanent database by default.

If NFS is active and you specify this qualifer, the changes are made to the permanent database only. For these changes to take effect, you must restart NFS.

/HOST=(host_name[,...])
Specifies the ULTRIX hosts from which access is to be given.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

If you specify /HOST=*, the file system is accessible from all hosts.

/UID=uid
Specifies the ULTRIX UID for the client that is to be given access.

Examples

1
```
UCX> ADD PROXY NFS_USER/GID=10/UID=10/HOST="june"
```

Gives the NFS client with a UID of 10 and GID of 10 the VMS identity associated with the account NFS_USER on host "june".

2
```
UCX> ADD PROXY NFS_USER/GID=10/UID=10/HOST=*
```

Gives the NFS client with a UID of 10 and GID of 10 from all hosts the VMS identity associated with the account WORK.

3
```
UCX> ADD PROXY NFS_USER/GID=10/UID=10/HOST=("may","june","july")
```

Gives the NFS client with a UID of 10 and GID of 10 the VMS identity associated with the account NFS_USER on hosts "may", "june", and "july".

ANALYZE CONTAINER

Verifies the integrity of an ULTRIX file system.

Format

ANALYZE CONTAINER "ultrix_logical_filesystem"

Command Qualifiers	Defaults
/[NO]CONFIRM	/NOCONFIRM
/[NO]REPAIR	/NOREPAIR

Restrictions

Before you can use the ANALYZE CONTAINER command, the underlying VMS file system name (that is, the VMS volume in which the container directory is located) must be bound to a logical file system with a BIND command.

It is recommended that you unbind the ULTRIX logical file system to prevent access to it while you issue this command.

This operation requires BYPASS privilege.

Parameter

"ultrix_logical_filesystem"
A device and directory containing an ULTRIX file system.

Description

The ANALYZE CONTAINER command is used to verify the integrity of an ULTRIX file system. It reports and optionally corrects problems within the structure of the container file representing the ULTRIX logical file system. The VMS volume on which the ULTRIX logical file system resides must be bound before you issue the ANALYZE CONTAINER command.

ANALYZE CONTAINER does the following:

- Superblock validation
- Inode validation
- Directory validation

- Internal allocation validation
- Directory hierarchy validation

Command Qualifiers

/[NO]CONFIRM

The CONFIRM qualifier is used with the /REPAIR qualifier and enables you to determine the course of action for each problem that needs repair.

When a problem is encountered by the ANALYZE CONTAINER command, a statement of the problem is displayed, along with the solution for that problem. You are then prompted for some action. Enter Y to repair the problem, N to ignore the problem, or G to change to NO CONFIRMATION mode (no further confirmation is requested). An example confirmation request is:

```
%UCX$-E-ANA_SUP_BADIICGSIZE, Problem: Bad initial inode cell
group size: <bad value>
        Solution: Will be replaced by default size (<good value>)
        CONFIRM [Y/N/G]:
```

The default is /NOCONFIRM.

/[NO]REPAIR

Controls whether errors found during the analysis are repaired.

The default is /NOREPAIR.

Example

```
UCX> BIND DUA0: "/dua0"
UCX> ANALYZE CONTAINER DUA0:[WORK1]
```

The integrity of the ULTRIX file system DUA0:[WORK1] is verified, but is not repaired if there is a problem, because the /REPAIR qualifier was not specified.

BIND

Makes a VMS or ULTRIX file system known as a Connection file system.

Format

BIND logical_filesystem "filesystem_name"

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires SYSPRV privilege and may require BYPASS privilege.

Parameters

logical_filesystem
The logical file system is one of the following:

- A VMS device name representing a VMS file system
- A device and directory containing an ULTRIX file system

"filesystem_name"
The ULTRIX-style name to be given to the logical file system. This is the file system name used by the NFS client in a mount operation.

Description

The BIND command makes a VMS or ULTRIX file system part of the Connection file system. This operation must be done on the VMS system before the NFS server can access the file system. The BIND command is logically equivalent to the VMS mount command.

If the logical file system is an ULTRIX file system (a file system that is held in a container file), the VMS disk the file system resides on must be bound before you can bind the logical file system.

Note *If the file system is to be available on all hosts of a VAXcluster system, then the file system must be bound on each host.*

Examples

1
```
UCX> BIND HOST$DUA1: "/DISK@VMS"
```

Declares a VMS device, HOST$DUA1:, to be a VMS-style Connection file system known as "DISK@VMS". File operations on any file within this file system will adhere to the VMS file system rules.

2
```
UCX> BIND HOST$DUA1:[ULT_FILESYS] "/ruser"
```

Declares an ULTRIX file system container directory, HOST$DUA1:[ULT_FILESYS], to be an ULTRIX-style Connection file system known as "/rusewr". File operations on any file within this file system will adhere to the ULTRIX file system rules.

CONVERT/VMS HOST

Converts an ULTRIX-formatted */etc/hosts* file to a UCX$HOST database.

Format

CONVERT/VMS HOST [source_file_spec]

Command Qualifiers	Defaults
/LOG	None.
/UPCASE	None.
/[NO]YP_FORMAT	/YP_FORMAT

Restrictions

This operation requires read/write access to the UCX$HOST database and read access to the source file */etc/hosts*.

This operation requires exclusive use of the UCX$HOST database.

Parameter

source_file_spec
The name of the file to be converted to a UCX$HOST file. If you do not specify the *source_file_spec*, the default file specification is [ ]ETC.HOSTS.

Description

The CONVERT/VMS HOST command converts an ULTRIX-formatted */etc/hosts* file to a UCX$HOST database.

If you specify /YP_FORMAT, the converted file has the following format:

host_internet_address host_name host_alias_list

The */etc/hosts* file, which is served by YP, also has this format.

If you specify /NOYP_FORMAT, the converted file has the following format:

host_name host_internet_address host_alias_list

Each field is delimited by space or tab characters.

The host database is specified by the logical name UCX$HOST. If the logical name is not defined, the CONVERT/VMS HOST command looks for UCX$HOST.DAT in your current directory. You must create the database using the UCX command CREATE HOST, before you use the CONVERT/VMS HOST command

A single host can have more than one Internet address assigned to it. However, if an Internet address is assigned to more than one host, an error occurs.

If you are using a Berkeley Internet Name Domain (BIND) formatted database on an ULTRIX system, use the */etc/hosts* file that was used to create the BIND database.

Command Qualifiers

/LOG
Displays records as they are being processed.

/UPCASE
Specifies that an uppercased alias name will be created for each host.

/[NO]YP_FORMAT
Specifies the ULTRIX file to be in Yellow Pages (YP) format. Specify /NOYP_FORMAT for a BIND database.

The default is /YP_FORMAT.

Example

```
UCX> CONVERT/VMS HOST
```

Converts an ULTRIX-formatted host database */etc/hosts* into a VMS-formatted host database (SYS$SYSTEM:UCX$HOST.DAT).

CONVERT/VMS NETWORK

Converts an ULTRIX-formatted */etc/networks* file to a UCX$NETWORK database.

Format

CONVERT/VMS NETWORK [source_file_spec]

Command Qualifiers	Defaults
/LOG	None.
/UPCASE	None.
/[NO]YP_FORMAT	/YP_FORMAT

Restrictions

This operation requires read/write access to the UCX$NETWORK database and read access to source file */ect/networks*. This operation requires exclusive use of the UCX$NETWORK database.

Parameter

source_file_spec
The name of the file to be converted to a UCX$NETWORK database. If you do not specify the *source_file_spec*, the default file specification is [ ]ETC.NETWORKS.

Description

The CONVERT/ VMS NETWORK command converts an ULTRIX-formatted */etc/networks* file to a UCX$NETWORK database.

The converted file has the following format:

network_name network_number network_alias_list

Each field is delimited by space or tab characters.

The network database is specified by the logical name UCX$NETWORK. If the logical name is not defined, the CONVERT/VMS NETWORK command looks for UCX$NETWORK.DAT in your current directory. You must create the

database using the UCX CREATE NETWORK command, before you use the CONVERT/VMS NETWORK command.

If you are using a BIND-formatted database on an ULTRIX system, use the */etc/networks* file that was used to create the BIND database.

Command Qualifiers

/LOG
Displays records as they are being processed.

/UPCASE
Specifies that an uppercased alias name will be created for each network name.

/[NO]YP_FORMAT
Specifies the ULTRIX file to be in Yellow Pages (YP) format.

Specify /NOYP_FORMAT for a BIND database.

The default is YP_FORMAT.

Example

```
UCX> CONVERT/VMS NETWORK
```

Converts an ULTRIX-formatted network database (*/etc/networks*) into a VMS-formatted network database (UCX$NETWORK).

CONVERT/VMS PROXY

Converts an ULTRIX-formatted */etc/passwd* file to a UCX$PROXY database.

Format

CONVERT/VMS PROXY [source_file_spec]

Command Qualifier	Default
/LOG	None.

Restrictions

This operation requires read/write access to the UCX$PROXY database and read access to source file */etc/passwd*. This operation requires exclusive use of the UCX$PROXY database.

Parameter

source_file_spec
The name of file to be converted to a UCX$PROXY database. If you do not specify the *source_file_spec*, the default file spcification is [ ]ETC.PASSWD.

Description

The CONVERT/VMS PROXY command converts an ULTRIX-formatted */etc/passwd* file to a UCX$PROXY database.

The relevant fields in the converted file have the following format:

user_name::uid:gid:::

The proxy database is specified by the logical name UCX$PROXY. If the logical name is not defined, the CONVERT/VMS PROXY command looks for UCX$PROXY.DAT in your current directory. You must create the database using the UCX CREATE PROXY command before you use the CONVERT/VMS PROXY command.

Command Qualifier

/LOG
Displays records as they are being processed.

Example

```
UCX> CONVERT/VMS PROXY
```

Converts an ULTRIX-formatted proxy database */etc/passwd* to a VMS-formatted proxy database (UCX$PROXY).

CONVERT/ULTRIX HOST

Converts a UCX$HOST database to an ULTRIX-formatted */etc/hosts* file.

Format

CONVERT/ULTRIX HOST [dest_file_spec]

Command Qualifiers	Defaults
/BYADDRESS	None.
/LOG	None.
/[NO]YP_FORMAT	/YP_FORMAT

Restrictions

This operation requires read access to the UCX$HOST database and write access to the destination file (*/etc/hosts*).

Parameter

dest_file_spec
The name of the destination file that is converted from the UCX$HOST database. If you do not specify the *source_file_spec*, the default file specification is [ ]ETC.HOSTS.

Description

The CONVERT/ULTRIX HOST command converts a UCX$HOST database to an ULTRIX-formatted */etc/hosts* file.

If you specify /YP_FORMAT, the destination file has the following format:

host_internet_address host_name host_alias_list

If you specify /NOYP_FORMAT, the destination file is formatted as follows:

host_name host_internet_address host_alias_list

The host database is specified by the logical name UCX$HOST. If the logical name is not defined, the CONVERT/ULTRIX HOST command looks for UCX$HOST in your current directory. You must create the database using the UCX CREATE HOST command, before you use the CONVERT/ULTRIX HOST command.

If you will be using the file to update a BIND-formatted database on an ULTRIX system, you need to keep the resulting */etc/hosts* formatted file as input to your BIND conversion on the ULTRIX system.

Command Qualifiers

/BYADDRESS
Specifies that converted file will be sorted by Internet address.

/LOG
Displays records as they are being processed.

/[NO]YP_FORMAT
Specifies the ULTRIX file to be in Yellow Pages (YP) format.

The default is YP_FORMAT.

If you will be using a BIND-formatted database on your ULTRIX system, use the /NOYP_FORMAT qualifier.

Example

```
UCX> CONVERT/ULTRIX HOST
```

Converts a VMS-formatted UCX$HOST database to an ULTRIX-formatted host database (*/etc/hosts*).

CONVERT/ULTRIX NETWORK

Converts a UCX$NETWORK database to an ULTRIX-formatted */etc/networks* file.

Format

CONVERT/ULTRIX NETWORK [dest_file_spec]

Command Qualifiers	Defaults
/BYADDRESS	None.
/LOG	None.

Restrictions

This operation requires read access to the UCX$HOST database and write access to the destination file (*/etc/network*).

Parameters

dest_file_spec
The default file is [ ]ETC.NETWORKS, which is an ULTRIX-formatted network file.

Description

The CONVERT/ULTRIX NETWORK command converts a UCX$NETWORK database to an ULTRIX-formatted */etc/networks* file.

The file format is:

network_name network_number network_alias_list

The network database is specified by the logical name UCX$NETWORK. If the logical name is not defined, the CONVERT/ULTRIX network command looks for UCX$NETWORK in your current directory. You must create the database using the UCX CREATE network command, before you use the CONVERT/ULTRIX NETWORK command.

If you will be using the file to update a BIND-formatted database on an ULTRIX system, you need to keep the resulting */ect/networks* formatted file as input to your BIND conversion on the ULTRIX system.

Command Qualifiers

/BYADDRESS
Specifies that converted file will be sorted by network number.

/LOG
Displays records as they are being processed.

Example

```
UCX> CONVERT/ULTRIX NETWORK/LOG
```

Converts a VMS-formatted UCX$NETWORK database to an ULTRIX-formatted network database (*/etc/networks*) and displays records as they are being processed.

CREATE CONTAINER

Creates a logical ULTRIX file system with an empty root directory.

Format

CREATE CONTAINER ultrix_logical_filesystem

Command Qualifiers	Defaults
/HOST=host_name	None.
/[NO]LOG	See text.
/OWNER=[uic]	See text.
/ROOT_MODE=mode	/ROOT_MODE=755
/SIZE=(option_list)	See text.
/UID=ultrix_uid	None.
/USER_NAME=vms_username	None.

Restrictions

This operation requires read/write access to the directory and SYSPRV or BYPASS privileges.

Parameter

ultrix_logical_filesystem
A device and directory that will contain an ULTRIX file system.

Description

The CREATE CONTAINER command creates a logical ULTRIX file system with an empty root directory. The file system consists of a VMS directory that contains a container file and an empty VMS directory corresponding to the ULTRIX root directory.

A container file is an RMS file whose name is the same as the container directory and whose type is .CONTAINER.

The protection for file access includes read (r), write (w), and execute (e). Directory files use the execute protection to allow searching of the directory.

Command Qualifiers

/HOST=host_name
If the proxy file contains multiple entries with the same user name and UID, the /HOST qualifier selects a specific proxy record or records.

/[NO]LOG
Determines whether information is displayed describing the specified UCX$PROXY database record to determine ownership. (Refer to /USER_NAME for a description of root directory ownership.)

The default is to display the values for the host, UID, GID, and user name from the selected UCX$PROXY database record.

/OWNER=[uic]
Specifies the VMS ownership of the container file directory and the container file itself. Other files in the container file directory are owned by the VMS users whose proxy file entries correspond with the ULTRIX owner UIDs of the individual files.

The default is the UIC in the UCX$PROXY database that corresponds to a UID of 0 and a GID of 1. This occurs because you must have SYSPRV or BYPASS privileges to use this command.

/ROOT_MODE=mode
Specifies the ULTRIX protection of the default system files (root directory, bitmap, and superblock file). This value is in octal. You must specify a protection for user, group, and others in that order. You specify the protection for each as a value. The values are as follows:

- 0 — No access
- 1 — Execute access only
- 2 — Write access only
- 3 — Write and execute access
- 4 — Read access only
- 5 — Read and execute access
- 6 — Read and write access
- 7 — Read, write, and execute

For example, /ROOT_MODE=751 would provide the following access:

- User — Read, write, and execute (7)
- Group — Read and execute access (5)
- Other — Execute access only (1)

The default is /ROOT_MODE=755, which provides the following protection for the owner, group, and world respectively: rwxr-xr-x.

/SIZE=(option_list)
Specifies various container file size attributes:

- INITIAL is the initial size, in VMS system blocks, of the container file. The default container file size is 39 blocks.
- EXTEND specifies the number of blocks by which the container file is to be extended when extension is necessary. The default is 160 blocks.
- MAXIMUM specifies a maximum size for the container file. Once the file reaches this size in blocks, no new files can be added without deleting old files. A MAXIMUM of 0 allows the container file to grow without limits. The default is 0.

/UID=ultrix_uid
Specifies an entry in the UCX$PROXY database to determine the ownership of the ULTRIX file system root directory.

The UID and GID fields in the UCX$PROXY entry are used to establish the root directory's ULTRIX identity, while the user name field is used to provide the VMS ownership.

The /HOST qualifier may be required if you want to access an entry in the UCX$PROXY database that does not have a unique UID and user name. For example, the same UID and user name combination could appear on multiple hosts.

You can use the /UID qualifier in any combination with the /HOST and /USER_NAME qualifiers.

/USER_NAME=vms_username
Selects the UID and GID from the proxy record that will own the root directory of the container file.

If you do not specify the /USER_NAME qualifier, the proxy record with a GID of 1 and a UID of 0 is selected by default. If there is no proxy entry for the UID of 0, the create container operation will fail.

Example

```
UCX> CREATE CONTAINER WORK1$:[SMITH]
```

Creates a container file WORK1$:[SMITH], which by default is owned by the VMS account registered for a client with a UID of 0 and a GID of 1 and has the default protection of 755.

If the /USER_NAME for the UID of 0 and GID of 1 (in the proxy database) is SYSTEM, then the container file and its root directory will be owned by SYSTEM.

CREATE DIRECTORY

Creates a directory in an ULTRIX file system.

Format

CREATE DIRECTORY "/ultrix_directory_spec"

Command Qualifiers	Defaults
/HOST=host_name	None.
/[NO]LOG	See text.
/MODE=mode_type	/MODE=755
/UID=ultrix_uid	None.
/USER_NAME=vms_username	None.

Restrictions

This operation requires read/write access to the directory. SYSPRV or BYPASS privileges are required if you use the /USER_NAME qualifier and specify a user name other than your own.

The file system must be bound.

Parameter

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name, zero or more directory names separated by slashes, and followed by the directory name to be created.

Description

The CREATE DIRECTORY command creates a directory in an ULTRIX file system.

Command Qualifiers

/HOST=host_name
If the proxy file contains multiple entries with the same user name and UID, the /HOST qualifier selects a specific proxy record or records.

/[NO]LOG
Determines whether information is displayed describing the specified UCX$PROXY database record to determine ownership. (Refer to /USER_NAME for a description of root directory ownership.)

The default is to display the values for the host, UID, GID, and user name from the specified UCX$PROXY database record.

/MODE=mode_type
Specifies the ULTRIX protection to be associated with the new directory. You must specify a protection for user, group, and others in that order. You specify the protection for each as a value. The values are as follows:

- 0 — No access
- 1 — Execute access only
- 2 — Write access only
- 3 — Write and execute access
- 4 — Read access only
- 5 — Read and execute access
- 6 — Read and write access
- 7 — Read, write, and execute

For example, /ROOT_MODE=751 would provide the following access:

- User — Read, write, and execute (7)
- Group — Read and execute access (5)
- Other — Execute access only (1)

The default is /ROOT_MODE=755, which provides the following protection for the owner, group, and world respectively: rwxr-xr-x.

/UID=ultrix_uid
Specifies an entry in the UCX$PROXY database to determine the ownership of the ULTRIX file system root directory.

The UID and GID fields in the UCX$PROXY entry are used to establish the root directory's ULTRIX identity, while the user name field is used to provide the VMS ownership.

The /HOST qualifier may be required if you want to access an entry in the UCX$PROXY database that does not have a unique UID and user name. For example, the same UID and user name combination could appear on multiple hosts.

You can use the /UID qualifier in any combination with the /HOST and /USER_NAME qualifiers. However, if you do not have SYSPRV or BYPASS privileges, the values you specify must correspond to the values for your user name in the PROXY database, or you will receive an error.

/USER_NAME=vms_username

Selects an entry in the proxy file. From the selected entry information, the UID, GID, and VMS UIC for the directory files are created.

If you have SYSPRV or BYPASS privileges and do not specify the /USER_NAME qualifier, the proxy record with a UID of 0 and a GID of 1 is selected by default.

SYSPRV or BYPASS privileges are required to select a user name that has a UIC that is different from the UIC of the process running UCX$UCP.EXE.

You can use the /USER_NAME qualifier in any combination with the /HOST and /UID qualifiers. However, if you do not have SYSPRV or BYPASS privileges, the values you specify must correspond to the values for your user name in the PROXY database, or you will receive an error.

Example

```
UCX> CREATE DIRECTORY "/user/smith"
```

Before the directory is created for the user "smith", the following BIND command would be issued:

```
BIND dua0:[group_a] "/user"
```

In these examples, the user "SMITH" is running UCX$UCP.EXE from the home directory ([SMITH]). The UIC for [SMITH] is [340,6] and the UCX$PROXY entry is defined as follows:

USER	UID	GID	HOST
SMITH	300	12	*
SYSTEM	0	1	*

If SMITH does not have SYSPRV or BYPASS privileges, the directory is created as follows:

UID = 300
GID = 12
UIC = [340,6]

If SMITH has SYSPRV of BYPASS privileges, the directory is created as follows:

UID = 0
GID = 1
UIC = [SYSTEM]

CREATE EXPORT

Creates an empty NFS export database.

Format

CREATE EXPORT

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires write access to the directory that contains the export database.

Description

The CREATE EXPORT command creates an empty NFS export database. If the UCX$EXPORT logical name is defined, it is used to specify the directory and file name for the database. If UCX$EXPORT is not defined, the database is created as [ ]UCX$EXPORT.DAT.

The UCX$EXPORT is created with the following protection:

- System — RWED
- Owner — RWED
- Group — RE
- World — RE

Example

```
UCX> CREATE EXPORT
```

Creates an empty NFS export database.

CREATE HOST

Creates an Internet host database.

Format

CREATE HOST

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires write access to the directory that contains the host database and read/write access to the database.

Description

The CREATE HOST command creates an Internet host database. This database will have one entry in it for LOCALHOST with an alias of *localhost*, at address 127.0.0.1. If the UCX$HOST logical name is defined, it is used to specify the directory and file name for the database. If UCX$HOST is not defined, the database is created as [ ]UCX$HOST.DAT.

The UCX$HOST is created with the following protection:

- System — RWED
- Owner — RWED
- Group — RE
- World — RE

Example

```
UCX> CREATE HOST
```

Creates an Internet host database.

CREATE NETWORK

Creates an empty network database.

Format

CREATE NETWORK

Command Qualifiers

None.

Defaults

None.

Restrictions

This operation requires write access to the directory that will contain the network database.

Description

The CREATE NETWORK command creates an empty network database. If the UCX$NETWORK logical name is defined, it is used to specify the directory and file name for the database. If UCX$NETWORK is not defined, the database is created as [ ]UCX$NETWORK.DAT.

The UCX$NETWORK is created with the following protection:

- System — RWED
- Owner — RWED
- Group — RE
- World — RE

Example

```
UCX> CREATE NETWORK
```

Creates an empty network database.

CREATE PROXY

Creates an empty NFS proxy database.

Format

CREATE PROXY

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires write access to the directory that contains the proxy database.

Description

The CREATE PROXY command creates an empty NFS proxy database. If the UCX$PROXY logical name is defined, it is used to specify the directory and file name for the database. If UCX$PROXY is not defined, the database is created as [ ]UCX$PROXY.DAT.

The UCX$PROXY is created with the following protection:

- System — RWED
- Owner — RWED
- Group — RE
- World — RE

Example

```
UCX> CREATE PROXY
```

Creates an empty NFS proxy database.

CREATE ROUTE

Creates an empty Internet route database.

Format

CREATE ROUTE

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires write access to the directory that contains the route database.

Description

The CREATE ROUTE command creates an empty Internet route database. The on-disk route database is referred to as the permanent database.

If the UCX$ROUTE logical name is defined, it is used to specify the directory and file name for the database. If UCX$ROUTE is not defined, the database is created as [ ]UCX$ROUTE.DAT.

The UCX$ROUTE is created with the following protection:

- System — RWED
- Owner — RWED
- Group — RE
- World — RE

Example

```
UCX> CREATE ROUTE
```

Creates an empty Internet route database.

DELETE CONTAINER

Deletes a Connection ULTRIX file system.

Format

DELETE CONTAINER ultrix_logical_filesystem

Command Qualifiers	Defaults
None.	None.

Restrictions

You cannot use wildcards in the ULTRIX logical file system name.

This operation requires read/delete access to the directory and may require BYPASS privilege.

Parameter

ultrix_logical_filesystem
A device and directory containing an ULTRIX file system.

The container file has a file type of .CONTAINER.

The container file specification cannot contain any wildcards.

Description

Deletes an ULTRIX file system by deleting the VMS container directory and any associated VMS files. The file system cannot be bound.

Example

```
UCX> DELETE CONTAINER WORK1$:[SMITH.ULTRIX_WORK]
```

Deletes the container file WORK1$:[SMITH.ULTRIX_WORK].

DIRECTORY

Lists information on one or more files in a Connection file system.

Format

DIRECTORY "/ultrix_directory_spec"

Command Qualifiers	Defaults
/FULL	See text.
/VMS	None.

Restrictions

This operation requires read/write access to the directory and may require BYPASS privilege.

Parameter

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name and zero or more directory names separated by slashes.

Description

The DIRECTORY command displays the list of files and provides information about a file or group of files in a Connection file system.

Command Qualifiers

/FULL
Specifies a comprehensive list of information that is to be provided for each file displayed by the DIRECTORY command. The /FULL qualifier includes the VMS file name.

The default is to provide a brief listing of the files in the dirctory.

/VMS
Provides the corresponding VMS file name for each file in the directory.

Examples

1

```
UCX> DIRECTORY/Full "/test_container"

Directory:  /test_container

 .
 VMS file: _$1$DISK:[SYSTEM.TESTCONTA]00012301$BFS.DIR;1
 Size                                File ID:    74497
  Blocks:            4               Owner
  Bytes:          1915                UID:          0
 Created:   1-AUG-1989 13:17:18.91    GID:          1
 Revised:   1-AUG-1989 13:17:19.24   Mode:        755  Type: Directory
 Accessed:  1-AUG-1989 13:16:20.52   Links:         2

 ..
 VMS file: _$1$DISK:[SYSTEM.TESTCONTA]00012301$BFS.DIR;1
 Size                                File ID:    74497
  Blocks:            4               Owner
  Bytes:          1915                UID:          0
 Created:   1-AUG-1989 13:17:18.91    GID:          1
 Revised:   1-AUG-1989 13:17:19.24   Mode:        755  Type: Directory
 Accessed:  1-AUG-1989 13:16:20.52   Links:         2

 .SUPER.SYS
 VMS file: no corresponding file
 Size                                File ID:     6145
  Blocks:            1               Owner
  Bytes:            54                UID:          0
 Created:   1-AUG-1989 13:17:18.91    GID:          1
 Revised:   1-AUG-1989 13:17:17.24   Mode:        644  Type: File
 Accessed:  1-AUG-1989 13:16:18.52   Links:         1

 .BITMAP.SYS
 VMS file: no corresponding file
 Size                                File ID:     6657
  Blocks:           16               Owner
  Bytes:          8187                UID:          0
 Created:   1-AUG-1989 13:17:18.91    GID:          1
 Revised:   1-AUG-1989 13:17:17.24   Mode:        644  Type: File
 Accessed:  1-AUG-1989 13:16:18.52   Links:         1
```

```
.HISTORY.SYS
VMS file: no corresponding file
Size                                    File ID:   66305
 Blocks:            1                   Owner
 Bytes:           129                    UID:          0
Created:   1-AUG-1989 13:17:18.91        GID:          1
Revised:   1-AUG-1989 13:17:17.24       Mode:        644  Type: File
Accessed:  1-AUG-1989 13:16:18.52       Links:         1
```

Displays a full directory listing of the file system "/test_container".

2 `UCX> DIRECTORY "/smith/work/plans"`

Displays the files that are contained in the directory "/smith/work/plans".

EXPORT

Copies a specified file within a Connection ULTRIX file system to a VMS file.

Format

EXPORT "ultrix_file_spec" vms_file_spec

Command Qualifiers	Defaults
None.	None.

Restrictions

You cannot use wildcards in either the ULTRIX or the VMS file specification.

Parameters

"ultrix_file_spec"
Begins with a slash (/), followed by the file system name, zero or more directory names separated by slashes, and optionally a file name. It cannot contain wildcards.

vms_file_spec
A valid VMS file specification without a node specification. It cannot contain wildcards.

Description

The EXPORT command copies a file within an ULTRIX file system to a VMS file.

Example

```
UCX> EXPORT "/smith/work/plans" WORK1$:[JONES]PLANS.TXT
```

Copies the file "plans" from the directory "/smith/work" into the VMS file PLANS.TXT in directory [JONES] on disk WORK1$.

IMPORT

Copies a VMS file to a file within a Connection ULTRIX file system.

Format

IMPORT vms_file_spec "ultrix_file_spec"

Command Qualifiers	Defaults
/[NO]CONVERT	/CONVERT
/HOST=host_name	None.
/[NO]LOG	/NOLOG
/MODE=mode_type	/MODE=755
/UID=ultrix_uid	None.
/USER_NAME=vms_username	None.

Restrictions

You cannot use wildcards in either the VMS or the ULTRIX file specification.

Parameters

vms_file_spec
A valid VMS file specification without a node specification. It cannot contain wildcards.

"ultrix_file_spec"
Begins with a slash (/), followed by the file system name, zero or more directory names separated by slashes, and optionally a file name. It cannot contain wildcards.

Description

The IMPORT command copies a VMS file to a file within an ULTRIX file system.

Command Qualifiers

/[NO]CONVERT
ULTRIX files are stored as STREAM_LF files on the VMS server. The /CONVERT qualifier is used to convert VMS record files to STREAM_LF files. If the /NOCONVERT qualifier is specified, no conversion is provided.

/HOST=host_name
If the proxy file has multiple entries with the same user name and UID, the /HOST qualifier selects a specific proxy record or records.

/[NO]LOG
Determines whether information is displayed describing the specified UCX$PROXY database record to determine ownership. (Refer to /USER_NAME for a description of root directory ownership.)

The default is to display the values for the host, UID, GID, and user name from the specified UCX$PROXY database record.

/MODE=mode_type
Specifies the ULTRIX protection to be associated with the new directory. You must specify a protection for user, group, and others in that order. You specify the protection for each as a value. The values are as follows:

- 0 — No access
- 1 — Execute access only
- 2 — Write access only
- 3 — Write and execute access
- 4 — Read access only
- 5 — Read and execute access
- 6 — Read and write access
- 7 — Read, write, and execute

For example, /ROOT_MODE=751 would provide the following access:

- User — Read, write, and execute (7)
- Group — Read and execute access (5)
- Other — Execute access only (1)

The default is /ROOT_MODE=755, which provides the following protection for the owner, group, and world respectively: rwxr-xr-x.

/UID=ultrix_uid
Selects an entry in the UCX$PROXY database to determine the ownership of the ULTRIX file.

The UID and GID fields in the UCX$PROXY entry are used to establish the file's ULTRIX identity, while the user name field is used to provide the VMS ownership.

The /HOST qualifier may be required if you want to access an entry in the UCX$PROXY database that does not have a unique UID and user name. For example, the same UID and user name combination could appear on multiple hosts.

You can use the /UID qualifier in any combination with the /HOST and /USER_NAME qualifiers. However, if you do not have SYSPRV or BYPASS privileges, the values you specify must correspond to the values for your user name in the PROXY database, or you will receive an error.

/USER_NAME=vms_username
Selects an in the UCX$PROXY database to determine the ownership of the ULTRIX file.

The UID and GID fields in the UCX$PROXY entry are used to establish the file's ULTRIX identity, while the user name field is used to provide VMS ownership.

The /HOST qualifier may be required if you want to access an entry in the UCX$PROXY database that does not have a unique UID and user name. For example, the same UID and user name combination could appear on multiple hosts.

If you have SYSPRV or BYPASS privileges and do not specify the /USER_NAME qualifier, the proxy record with a UID of 0 and a GID of 1 is selected by default.

SYSPRV or BYPASS privileges are required to select a user name that has a UIC that is different from the UIC of the process running UCX$UCP.EXE.

You can use the /USER_NAME qualifier in any combination with the /HOST and /UID qualifiers. However, if you do not have SYSPRV or BYPASS privileges, the values you specify must correspond to the values for your user name in the PROXY database, or you will receive an error. If you do not specify the /USER_NAME qualifier, the proxy record with a GID of 1 and a

UID of 0 is selected by default. If there is no proxy entry for the UID of 0, the import operation will fail.

Example

```
UCX> IMPORT WORK1$:[JONES]JOBS.TXT "/smith/work/jobs"
```

Copies the VMS file JOBS.TXT into the ULTRIX directory "/smith/work" with a file name of "jobs".

See the description of the CREATE DIRECTORY command for the way the file's ownership is determined.

LOOP

Tests the connectivity path to a specific host in the network by transmitting test data blocks to that host.

Format

LOOP [host_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/TIMEOUT=seconds	None.

Restrictions

This operation requires a privileged UIC, SYSPRV, or OPER privileges.

Parameter

host_name
The host name or Internet address of the host that is a source or destination of messages.

Each host name must have a unique Internet address.

Description

The LOOP command tests the connectivity path to a specific host in the network by transmitting test blocks of data to that host. This test is implemented by Internet Control Message Protocol (ICMP) ECHO and ECHOREPLY messages passed at the Internet Protocol (IP) level.

If no host name is specified in the command, loopback testing is done on the local host.

Command Qualifiers

/ADDRESS=internet_address
Specifies the target host by Internet address. The /ADDRESS qualifier provides an alternate way of specifying the target host.

You cannot specify both the *host_name* parameter and the /ADDRESS qualifier.

It is recommended that you use the *host_name* parameter instead of this qualifier.

/TIMEOUT=seconds
Specifies in seconds how long the LOOP command will be activated.

Examples

1
```
UCX> LOOP "moe" /TIMEOUT=25
```

Specifies that the local host test the connectivity path to host "moe" for a maximum of 25 seconds.

2
```
UCX> LOOP
```

Tests the local Internet software only; no packets are sent on the network.

REMOVE DIRECTORY

Removes a link to a directory in the specified file system.

Format

REMOVE DIRECTORY "/ultrix_directory_spec"

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires read/delete access to the directory and may require BYPASS privilege.

Parameter

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name and zero or more directory names separated by slashes.

Description

Removes a link to a directory in the named file system. If there are no other links to the directory, the directory is deleted.

Example

```
UCX> REMOVE DIRECTORY "/smith/work"
```

Removes a link to the directory "/smith/work".

REMOVE EXPORT

Deletes access to a Connection ULTRIX file system for one or more specified remote hosts.

Format

REMOVE EXPORT "/ultrix_directory_spec"

Command Qualifiers	Defaults
/[NO]CONFIRM	/CONFIRM
/HOST=(host_name[,...])	/HOST=*

Restrictions

This operation requires read/write access to the UCX$EXPORT database.

Parameter

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name and zero or more directory names separated by slashes.

Description

The REMOVE EXPORT command removes access to a Connection ULTRIX file system for one or more specified remote hosts.

To remove all access to the targeted file system, issue the REMOVE EXPORT command without the /HOST qualifier.

Command Qualifiers

/[NO]CONFIRM
Disables or enables confirmation before export information is removed during a wildcard delete operation. The default for wildcard delete operations is /CONFIRM.

/HOST=(host_name[,...])
Removes access to the ULTRIX file system for the specified remote host or hosts. If you specify the /HOST qualifier, you must specify a host name.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

If you do not specify the /HOST qualifier, access to all host is removed by default; therefore the default is the equivalent of /HOST=*.

Examples

1
```
UCX> REMOVE EXPORT "/smith/work" /HOST="curley"
```

The ULTRIX host "curley" can no longer access "/smith/work".

2
```
UCX> REMOVE EXPORT "/smith/*" /HOST=*
```

Deletes export entries for all file systems below "/smith" for all hosts.

REMOVE FILE

Removes a link to the file entry for the specified file specification.

Format

REMOVE FILE "ultrix_file_spec"

Command Qualifiers

None.

Defaults

None.

Restrictions

This operation requires read/delete access to the directory and may require BYPASS privilege.

Parameter

"ultrix_file_spec"

Begins with a slash (/), the file system name, zero or more directory names separated by slashes, and followed by a file name.

Description

The REMOVE FILE command removes a link to the file entry for the targeted file specification. If the file entry is the last link to the file, the file is deleted.

Example

```
UCX> REMOVE FILE "/smith/work/plans"
```

Removes a link to the file "plans".

REMOVE PROXY

Deletes the NFS proxy authorization for one or more ULTRIX clients.

Format

REMOVE PROXY [vms_account]

Command Qualifiers	Defaults
/GID=gid	/GID=*
/[NO]CONFIRM	/CONFIRM
/HOST=(host_name[,...])	/HOST=*
/UID=uid	/UID=*

Restrictions

This operation requires read/delete access to the UCX$PROXY database.

Parameter

vms_account
Specifies the VMS account on a VMS system.

Description

The REMOVE PROXY deletes the NFS proxy authorization for one or more ULTRIX clients. If NFS is active, any changes you make take effect immediately and affect both the permanent and volatile database.

Command Qualifiers

/GID=gid
Removes proxy authorization for the client having the specified GID. If you do not specify the /GID qualifier, proxy authorization is removed for all clients.

The default is /GID=*

/[NO]CONFIRM
Disables or enables confirmation before proxy information is removed during a wildcard delete operation. The default for wildcard delete operations is /CONFIRM.

/HOST=(host_name[,...])
Removes proxy authorization for the specified hosts.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

If you do not specify the /HOST qualifier, proxy authorization is removed from all hosts by default; therefore the default is the equivalent of /HOST=*.

/UID=uid
Removes proxy authorization for the client having the specified UID.

If you do not specify the /UID, proxy authorization is removed from all clients.

Examples

1
```
UCX> REMOVE PROXY "proxy1" /HOST=GOLDEN/UID=83
```
Removes authorization for client 83 on host GOLDEN from VMS account "proxy1".

2
```
UCX> REMOVE PROXY /HOST=GOLDEN/UID=83
```
Removes authorization for user 83 from host GOLDEN.

3
```
UCX> REMOVE PROXY /HOST=(GOLDEN,GATE)
```
Removes authorization for any client on hosts GOLDEN and GATE.

4
```
UCX> REMOVE PROXY /UID=83
```
Removes authorization for user 83 from any host.

SET ARP

Maps dynamically between DARPA Internet and Ethernet addresses.

Format

SET [NO]ARP ethernet_address [host_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/[NO]PERMANENT	/PERMANENT
/[NO]PUBLIC	/PUBLIC

Restrictions

This operation requires operator (OPER) privilege.

Parameters

ethernet_address

An Ethernet address is 48 bits in length. Ethernet addresses are represented by six pairs of hexadecimal digits (6 bytes), separated by hyphens (for example, AA-01-23-45-67-FF). The bytes are displayed from left to right in the order in which they are transmitted; bits within each byte are transmitted from right to left.

If your Internet network coexists with a DECnet network, the addresses of the DECnet nodes are translated into Ethernet addresses. For more information, refer to the *VMS Networking Manual*.

host_name

The name of the host that is a source or destination of ARP messages.

Each host name must have a unique Internet address defined in the host database.

Description

The Address Resolution Protocol (ARP) maps dynamically between DARPA Internet and 10Mb/s Ethernet addresses.

ARP caches Internet-Ethernet address mappings every 20 minutes, by default. When an interface requests a mapping for an address not in the cache, ARP queues the message that requires the mapping and broadcasts a message on the associated network that requested the address mapping. If a response is provided, the new mapping is cached and any pending messages are transmitted. An interface mapping request times out in 3 minutes.

You may want to statically provide mapping information under one of the following conditions:

- ARP is running and a change has been made to the network interface on a host. You would want to flush the Internet-Ethernet mapping tables to allow the new address change to be made known to the system.
- A host system you want to communicate with does not support ARP; therefore, you would statically map Internet addresses to Ethernet addresses.

The SET ARP command has two parameters: the Ethernet address of the targeted interface and the host name of where the targeted interface resides. The *host_name* parameter is optional. If you have not assigned a name to the host, the /ADDRESS qualifier must be used.

Use the SET NOARP command to remove an Internet address to Ethernet address mapping.

Command Qualifiers

/ADDRESS=internet_address
Specifies the Internet address of the host on which the targeted interface resides. You cannot specify both the *host_name* parameter and the /ADDRESS qualifier.

/[NO]PERMANENT
Specifies whether the ARP mapping is to be permanent for the host.

Use the /NOPERMANENT qualifier to have the ARP mapping removed after the default caching time interval (20 minutes).

The default is /PERMANENT.

/[NO]PUBLIC

Specifies whether the ARP code will respond to ARP requests from other hosts to the specified host.

Specify the /NOPUBLIC qualifier to define mapping information on the local host only.

The default is /PUBLIC.

Example

```
UCX> SET ARP AA-02-04-05-06-07 MYHOST
```

Sets the Ethernet address of host MYHOST to AA-02-04-05-06-07.

SET COMMUNICATION

Sets the Internet software parameters for communication on the local host.

Format

SET COMMUNICATION

Command Qualifiers	Defaults
/[NO]BROADCAST	/NOBROADCAST
/CHECKSUM=option_list	See text.
/CLUSTER_TIMER=number	/CLUSTER_TIMER=0
/DEVICE_SOCKETS=number	/DEVICE_SOCKETS=30
/[NO]FORWARD	/NOFORWARD
/IRP=option_list	See text.
/LARGE_BUFFERS=option_list	See text.
/[NO]LOCAL_HOST=host_name	None.
/REASSEMBLY_TIMER=number	/REASSEMBLY_TIMER=15
/SMALL_BUFFERS=option_list	See text.
/TCP_QUOTA=option_list	See text.
/UDP_QUOTA=option_list	See text.

Restrictions

This operation requires operator (OPER) privilege.

Description

The SET COMMUNICATION command sets the Internet software parameters for communication on the local host.

Command Qualifiers

/[NO]BROADCAST

Enables or disables privilege checking for broadcast messages sent from the local host.

The /BROADCAST qualifier enables nonprivileged users to send broadcast messages.

The /NOBROADCAST qualifier specifies that the user must have a privileged UIC, SYSPRV, BYPASS, or OPER privileges to send broadcast messages.

The default is /NOBROADCAST

DECrpc applications use broadcast messages to locate servers. If DECrpc is being used on your system, you may want to disable the privilege checking.

/CHECKSUM=option_list
Enables and disables checksum validation of messages. The option determines at what protocol level the checksum validation is performed. The checksum is a 16-bit value calculated as the one's complement of the sum of the one's complement of consecutive 16 bits of data. The options are as follows:

Option	Default	Meaning
[NO]IP	IP	IP level (checksum validation of the IP header)
[NO]TCP	TCP	TCP level (checksum validation of the TCP header and data)
[NO]UDP	UDP	UDP level (checksum validation of the UDP header and data)

/CLUSTER_TIMER=number
Specifies the maximum number of minutes in which a particular host will respond to the cluster host name. The default value is 0, meaning that the host will hold the cluster host name until the Internet software is stopped or the interface is deleted.

/DEVICE_SOCKETS=number
Specifies the maximum number of device-sockets. The default value is 30. Refer to the SHOW COMMUNICATION command for information on tuning this number. Also, refer to Chapter 3 for information on how tuning this number can impact the nonpaged pool.

/[NO]FORWARD
Specifies whether IP messages are to be forwarded. The default is /NOFORWARD.

/IRP=option_list
Specifies the quota for the Internet I/O request packets (IRPs).

The options for setting the IRP quota are:

- MAX:number — Maximum number of IRPs (that is, the IRP quota). The default is 200 IRPs.
- FREE:number — Maximum number of IRPs in the free list. The free list is a lookaside queue of free Internet packets, preallocated from the VMS nonpaged pool. The default is 20 IRPs.

It is recommended that the number of free IRPs exceed the IRP quota, and that the IRP quota not exceed the number of IRPs defined with the VMS system parameter IRPCOUNT.

Refer to the SHOW COMMUNICATION/MEMORY command for information on tuning the IRP quotas.

/LARGE_BUFFERS=option_list

Specifies the number of static and dynamic buffers. The Internet software maintains two types of large buffers: static and dynamic. Static buffers are allocated when the Internet software is started and deallocated when a STOP COMMUNICATION command is issued. Dynamic buffers are additional buffers that the software may require to hold messages before the messages can be processed. The number of buffers used depends on the Internet software load. If not enough buffers are specified, messages are dropped at peak loads.

The options for setting large buffers are:

- MIN:number — Number of static buffers.
- MAX:number — Total number of static and dynamic buffers.

  The minimum maximum number for large buffers is 15. Note, you cannot change the minimum for the large buffers are if the Internet has been started.

See Chapter 3 for information on the impact of this number on the nonpaged pool.

/[NO]LOCAL_HOST=host_name

Specifies the local host name.

The /NOLOCAL_HOST qualifier deletes the local host name definition. You cannot specify a *host_name* with /NOLOCAL.

/REASSEMBLY_TIMER=number
Specifies the maximum time spent reassembling an IP fragmented datagram received from the network. The default value is 15 seconds. Refer to the SHOW COMMUNICATION/MEMORY command for information on tuning this number.

/SMALL_BUFFERS=option_list
Specifies the number of static and dynamic buffers. The Internet software maintains two types of small buffers: static and dynamic. Static buffers are allocated when the Internet software is started and deallocated when a STOP COMMUNICATION command is issued. Dynamic buffers are additional buffers that the software may require to hold messages before the messages can be processed. The number of buffers used depends on the Internet software load. If you do not specify enough buffers, messages will be dropped at peak loads.

Typically, you would allocate three times more small buffers than large buffers.

The options for setting small buffers are:

- MIN:number — Number of small static buffers.

  The maximum for the minimum number of small buffers is 127.
- MAX:number — Total number of small static and dynamic buffers.

Note, you cannot change the minimum for the large buffers are if the Internet has been started. See Chapter 3 for information on the impact of this number on the nonpaged pool.

/TCP_QUOTA=option_list
Specifies the queue size (in bytes) for TCP messages.

The options for setting TCP message queue size are:

- RECEIVE:number — Receive queue size. The default value is 4096 bytes.
- SEND:number — Send queue size. The default value is 4096 bytes.

/UDP_QUOTA=option_list
Specifies the queue size (in bytes) for UDP messages.

The options for setting UDP message queue size are:

- RECEIVE:number — Receive queue size. The default value is 9000 bytes.

- SEND:number — Send queue size. The default value is 9000 bytes.

Refer to the SHOW PROTOCOL UDP command for information on tuning these quotas.

Examples

1
```
UCX> SET COMMUNICATION/LARGE=(MIN=10, MAX=30)
```

Sets the number of static buffers to 10 and the total number of static and dynamic buffers to 30.

2
```
UCX> SET COMMUNICATION/CHECKSUM=(NOIP,NOTCP,NOUDP)
```

Disables the checksum validation for IP, TCP, and UDP protocols.

SET HOST

Defines or modifies the UCX$HOST database on the local host.

Format

SET [NO]HOST host_name

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
[NO]ALIAS=[host_name_list]	See text.
/[NO]CONFIRM	See text.

Restrictions

This operation requires read/write/delete access to the UCX$HOST database.

Parameter

host_name
The name of the host that is a source or destination of Internet communications.

Each host name must have a unique Internet address defined in the host database.

You cannot delete a host by specifying an alias name.

Description

The SET HOST command defines or modifies the UCX$HOST database on the local host. It provides maintenance for both the local and the remote hosts. In the ULTRIX environment, it is the equivalent of maintaining the */etc/hosts* file. The Connection supports hosts with more than one Internet address.

Command Qualifiers

/ADDRESS=internet_address
Defines the Internet address of a particular host.

/[NO]ALIAS=[host_name_list]
Provides a means of referring to a host by several different host names. It is used for convenience. This qualifier is not available for the SET NOHOST command.

Use the /NOALIAS qualifier to delete any or all aliases for a host.

/[NO]CONFIRM
Enables or disables confirmation before removal of a host record. The default is /CONFIRM.

Examples

1
```
UCX> SET HOST MOE /ADDRESS = 128.33.33.8 /ALIAS="moe"
```
Sets the Internet address of host MOE to 128.33.33.8 and establishes "moe" to be an alias for host MOE.

2
```
UCX> SET HOST MOE /ALIAS="MOE_2"
```
Establishes "MOE_2" as an alias for MOE.

3
```
UCX> SET HOST MOE /ADDRESS = 128.33.33.9
```
Establishes a second Internet address for MOE.

4
```
UCX> SET HOST MOE /ADDRESS = 128.33.33.9 /ALIAS="MOE_3"
```
Establishes "MOE_3" as an alias for MOE's second Internet address.

5
```
UCX> SET HOST MOE /NOALIAS="MOE_2"
```
Deletes "MOE_2" as an alias for MOE.

6
```
UCX> SET NOHOST MOE /NOCONFIRM
```
Deletes MOE and all of its aliases.

SET INTERFACE

Defines a communication interface residing on the local host.

Format

SET [NO]INTERFACE interface_name

Command Qualifiers	Defaults
/[NO]ACTIVE	/ACTIVE
/ADDRESS=internet_address	None.
/[NO]ARP	/ARP
/BROADCAST_MASK=internet_address	See text.
/C_ADDRESS=internet_address	See text.
/C_BROADCAST_MASK=internet_address	See text.
/C_NETWORK=internet_address	None.
/[NO]CLUSTER=host_name	See text.
/HOST=host_name	None.
/[NO]LOOPBACK	/NOLOOPBACK
/NETWORK_MASK=internet_address	See text.
/RECEIVE_BUFFERS=number	/RECEIVE_BUFFERS=10
/[NO]TRAILER	/NOTRAILER

Restrictions

This operation requires operator (OPER) privilege, read access to the UCX$HOST and UCX$NETWORK databases, and read/write/delete access to the UCX$ROUTE database.

Every host on the same network must have the same network_mask.

Parameters

interface_name
Either the name of an Internet interface or an Internet pseudo-interface.

An Internet interface has the following format:

- A 2-character string followed by a unit number. The unit number is 0 for the first Ethernet controller, 1 for the second Ethernet controller, and so on.

The following table shows the interface names and their corresponding Ethernet controllers:

Interface Name	Ethernet Controllers	VMS Device Names
qe	DEQNA, DELQA, DEQTA	XQ
se	DESVA	ES
ni	DEBNA, DEBNE	ET
de	DEUNA, DELUA	XE
xe	DEMNA	EX
ez	SEGEC	EZ

An Internet pseudo-interface has the following format:

- A 3-character name, followed by a maximum of 5 digits for the pseudo-interface unit number. The maximum unit number is 65535. The first two characters in the pseudo-interface name are the same as the two characters in the Internet interface name (such as, qe, de, se, and ni). The third character indentifies the Ethernet controller number that corresponds to the pseudo-interface, as follows:
  - a is Ethernet controller 0
  - b is Ethernet controller 1
  - c is Ethernet controller 2

    .
    .
    .

  The unit number is used to differentiate interfaces that have the same name.

Description

The SET INTERFACE command defines the communication interface residing on the local host by associating an Internet address with the interface name. This command can also modify the parameters of an existing interface.

The SET NOINTERFACE command removes a communication interface from the network. If you issue the SET [NO]INTERFACE command, a message is logged to the network operator's log file and displayed on the operator's console.

Command Qualifiers

/[NO]ACTIVE
Sets the state of the interface to active or inactive.

The default is /ACTIVE.

/ADDRESS=internet_address
Specifies the Internet address of the local host that will use this interface for Ethernet communications. This qualifier cannot be used with the /HOST qualifier.

Each interface must have a unique Internet address.

/[NO]ARP
When specified after interface creation and the interface has the ARP mechanism, it enables or disables Internet-Ethernet address mapping between network-level addresses and Internet-level addresses.

When specified at the time of interface creation, the Internet-Ethernet address mapping is available on the interface.

The default is /ARP.

For more information on ARP, see the SET ARP command.

/BROADCAST_MASK=internet_address
Makes the broadcast address known to the network interface. Whenever a message is transmitted to the broadcast address, every host on the network that specified the broadcasted address as its broadcast mask receives the message.

An Internet address consists of two fields: a network number and a host number. Depending on the type of address class, these fields vary in length. However, the maximum length of the ASCII string is 15 characters.

The Internet software calculates a default broadcast mask by using the network number field from the network mask, and by setting all bits in the host number field to 1.

/C_ADDRESS=internet_address
Specifies a single address for the VAXcluster, so that the whole cluster of homogeneous hosts appears as a single host to the remote Internet hosts. The address establishes an alias host identifier for the cluster.

If you use this qualifier, remote clients route messages to the cluster host identifier by way of one of the cluster members.

/C_BROADCAST_MASK=internet_address
Makes the cluster broadcast mask known to the system. For a more detailed discussion of how the default broadcast mask is calculated, refer to the description of the /BROADCAST_MASK qualifier.

/C_NETWORK=internet_address
Specifies the network mask of the cluster network. This mask is specific to the cluster host network. For a more detailed discussion of how the default network mask is calculated, refer to the description of the /NETWORK_MASK qualifier.

/[NO]CLUSTER=host_name
Specifies the cluster host name (the alias host identifier).

Before using the /CLUSTER=host_name qualifier, you must define the specified host name in the UCX$HOST database. The /CLUSTER=host_name qualifier is used to associate the alias host identifier to each interface within the VAXcluster system.

The /NOCLUSTER qualifier is used to disable Internet cluster processing on the specified interface. It does not take a value, and cannot be specified with any other qualifier.

Note *If you specify /NOCLUSTER, active communication is aborted for any applications that are bound to the cluster alias name.*

/HOST=host_name
Specifies the name of the local host that will be using the specified interface for Ethernet communications.

/[NO]LOOPBACK
Specifies that the interface is to perform in loopback mode.

The interface must be in the inactive state before you select the interface for loopback testing. After you select the interface for loopback testing, you activate the interface with the SET INTERFACE/ACTIVE command.

In loopback mode, packets are not sent to the network; instead, they are given directly to the receiver on the same host. The interface functions like a short circuit at the IP level.

The default is /NOLOOPBACK.

/NETWORK_MASK=internet_address
Specifies which part of the host field of an Internet address is to be interpreted as the subnet. An Internet address consists of two fields: a network number and a host number. Depending on the type of address class, these fields vary in length. The Internet software calculates a default network mask as follows:

- The bits representing the network fields are set to 1
- The bits representing the host field are set to 0

The host field can be further divided into a subnetwork and a host field; this definition is site specific. If you are using subnets, you must explicitly specify the network mask to include the subnet field; otherwise the network field will not include the subnet field.

/RECEIVE_BUFFERS=number
Specifies the number of the receive buffers that are preallocated and attached to the interface. Chose a number high enough to allow the longest datagrams to be received in the preallocated buffers. The default value is 10 buffers.

/[NO]TRAILER
When specified at the time of interface creation, the /TRAILER qualifier enables the use of trailer headers. Trailer headers are protocol headers that are attached at the end of the data packet instead of at the front of the data. By aligning data at memory pages and using the trailer headers, the speed of the transmitting and receiving packets is increased.

The /TRAILER qualifier enables the trailer link level encapsulation when receiving a message. If a network interface supports trailers and all the involved systems support trailer protocol, the system, when possible, encapsulates outgoing messages to minimize the number of memory-to-memory copy operations performed by the receiver.

Setting the /NOTRAILER qualifier disables the encapsulation.

The default is /NOTRAILER.

Example

```
UCX> SET INTERFACE qe0 /HOST=MYHOST /NETWORK_MASK=255.255.0.0-
_UCX> /BROADCAST_MASK=128.30.0.0 /ARP /TRAILER
```

Sets the network mask to 255.255.0.0, the broadcast mask to 128.30.0.0, and enables ARP and trailer headers for interface qe0 on host MYHOST.

SET NAME_SERVICE

Modifies the name service parameters.

Format

SET NAME_SERVICE

Command Qualifiers	Defaults
/DISABLE	None.
/[NO]DOMAIN=domain_name	See text.
/ENABLE	None.
/[NO]SERVER=(host_name[,...])	See text.
/SYSTEM	None.
/TRANSPORT=protocol_name	/TRANSPORT=UDP

Restrictions

If you use the /SYSTEM qualifier, SYSPRV or BYPASS privileges are requred.

Description

The SET NAME_SERVICE command modifies the name service parameters.

By default, only the process-specific name service parameters are modified. To modify the system-wide name service parameters, you must use the /SYSTEM qualifier.

Command Qualifiers

/DISABLE

Disables the BIND name service resolver. If you use this qualifier, there is no remote lookup of host information.

You must use this qualifier in conjunction with the /SYSTEM qualifier.

/[NO]DOMAIN=domain_name

Defines the default domain. The domain name is appended to all host name references made from the local process. Domain names with lowercase characters must be placed in quotation marks to preserve case sensitivy. For example:

```
UCX> SET NAME_SERVICE/DOMAIN="abc.xyz.com"
```

The /NODOMAIN qualifier deletes the process-specific definition of the domain. The /NODOMAIN qualifier cannot be used in conjunction with the /SYSTEM qualifier. There must be a system definition for the local domain.

/ENABLE

Enables the BIND service resolver. If you use this qualifier, remote database lookups are made if a request cannot be satisfied by the local database.

You must use this qualifier in conjunction with the /SYSTEM qualifier.

/[NO]SERVER=(host_name[,...])

Specifies the BIND servers known to the local host. If a server name list exists, the specified servers are added to the list. The host name list can include a maximum of three host names. The server name can be either a host name or Internet address.

The host name list is ordered by request preference. For example, the initial request for a host name or address is sent to the first host in the list. If that host is unavailable, the request is sent to the second host in the list, and so on.

If you specify more than one server name, you must separate them with commas and enclose them in parentheses. Also, to ensure that each server name is explicit, enclose each in quotation marks.

The /SERVER qualifier requires a value for the host name.

The /NOSERVER qualifier removes the specified servers. If the qualifier is specified with no value, then all server definitions are deleted.

You cannot specify the /NOSERVER qualifier without a value and the /SYSTEM qualifier. Also, you cannot remove all the server definitions from the system parameter; there must be at least one server definition.

/SYSTEM

Specifies that system-wide parameters are to be modified. This qualifier requires SYSPRV or BYPASS privileges.

If you do not specify this qualifier, then only the process-specific parameters are modified.

/TRANSPORT=protocol_name

Specifies the protocol to be used in communicating to the remote server. The supported protocols are UDP and TCP. The default is /TRANSPORT=UDP.

Examples

1 `UCX> SET NAME_SERVICE/ENABLE/SYSTEM`

Enables the BIND resolver.

2 `UCX> SET NAME_SERVICE/SERVER=WHY_ME/SYSTEM`

Defines the host WHY_ME as a BIND server system wide.

3 `UCX> SET NAME_SERVICE/SERVER="whyyou"/SYSTEM`

Defines the host "whyyou" as another BIND server system wide.

4 `UCX> SET NAME_SERVICE/SERVER=(WHY_OU, "why_es")/SYSTEM`

Defines the hosts WHY_OU and "why_es" as BIND servers system wide.

5 `UCX> SET NAME_SERVICE/SERVER=MY_OWN`

Defines the host MY_OWN as a process-specific BIND server. Requests from this process will not be sent to the system wide BIND servers.

6 `UCX> SET NAME_SERVICE/NOSERVER`

Removes all BIND servers for the process.

SET NETWORK

Assigns a network name to a network number.

Format

SET [NO]NETWORK network_name

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/[NO]ALIAS=(network_name[,...])	See text.
/[NO]CONFIRM	/CONFIRM

Restrictions

This operation requires read/write/delete access to the UCX$NETWORK database.

Parameter

network_name
An ASCII string that identifies a network number.

You cannot delete a network by specifying an alias name.

Description

The SET NETWORK command assigns a network name to a network number. In the ULTRIX environment, it is equivalent to maintaining the */etc/networks* file.

Command Qualifiers

/ADDRESS=internet_address
Specifies the Internet address for the named network.

/[NO]ALIAS=(network_name[,...])
Specifies a list of alias network names for a network.

If you specify more than one network name, you must separate them with commas and enclose them in parentheses. Also, to ensure that each network name is explicit, enclose each in quotation marks.

Use the /NOALIAS qualifier to delete all aliases for the specified networks.

/[NO]CONFIRM
Disables or enables confirmation before removal of a network record.

The default is /CONFIRM.

Examples

1
```
UCX> SET NETWORK MYNET /ADDRESS=128.30.30.10
/ALIAS=MYNET_1
```

Assigns the network name MYNET to Internet address 128.30.30.10 and establishes MYNET_1 as an alias for MYNET.

2
```
UCX> SET NETWORK MYNET /ALIAS=MYNET_2
```

Establishes a second alias, MYNET_2, for MYNET.

3
```
UCX> SET NETWORK MYNET /NOALIAS=MYNET_2
```

Deletes the MYNET_2 alias for MY NET.

4
```
UCX> SET NONETWORK MYNET
```

Deletes MYNET and all of its aliases.

SET NFS_SERVER

Modifies the characteristics of the NFS server.

Format

SET NFS_SERVER

Command Qualifiers	Defaults
/DISABLE=reporting_option	See text.
/ENABLE=reporting_option	See text.
/GID_DEFAULT=gid	/GID_DEFAULT= –2
/INACTIVITY_TIMER=time	None.
/THREADS=number	None.
/UID_DEFAULT=uid	/UID_DEFAULT= –2
/XID=number	None.

Restrictions

This operation requires SYSNAM, WORLD, and either SYSPRV or BYPASS privileges.

Description

The SET NFS_SERVER command modifies the characteristics of an NFS server. Some of these characteristics can be modified immediately (dynamically); other modifications do not take place until the server is restatrted (statically).

The following qualifiers modify the specified characteristic dynamically:

- /DISABLE
- /ENABLE
- GID_DEFAULT
- /INACTIVITY_TIMER
- UID_DEFAULT

The following qualifiers modify the specified characteristic dynamically:

- /THREADS
- /XID

You initialize the NFS server by running the UCX$NFS_STARTUP.COM procedure (located in SYS$MANAGER), which defines a set of logical names that provide the NFS server with default characteristics. The UCX$NFS_STARTUP.COM procedure is run automatically when you run UCX$STARTUP.COM. If you want to change the characteristics of the server permanently, modify the logical name values.

Command Qualifiers

/DISABLE=reporting_option
Disables reporting with one of the following options:

- ERROR — Disables error logging
- OPCOM — Disables reporting to OPCOM

By default, both reporting options are enabled.

/ENABLE=reporting_option
Enables reporting with one of the following options:

- ERROR — Enables error logging
- OPCOM — Enables reporting to OPCOM

By default, both reporting options are enabled.

GID_DEFAULT=gid
Specifies the ULTRIX default GID that is to be associated with a file that was not created through the NFS server.

The default is /GID_DEFAULT= –2.

/INACTIVITY_TIMER=time
Specifies the time in seconds that files will remain open if they are not accessed within the specified time. The format of the time is *mm:ss*, where *mm* specifies the number of minutes and *ss* specifies the number of seconds.

/THREADS=number
Specifies the maximum number of simultaneous requests that can be processed by the NFS server. This qualifier is static; you can modify it only by restarting the server.

/UID_DEFAULT=uid
Specifies the ULTRIX default UID that is to be associated with a file that was not created through the NFS server.

The default is /UID_DEFAULT= –2.

/XID=number
Specifies the maximum buffer allocation for XID caching. This qualifier is static; you can modify it only by restarting the server.

Example

```
UCX> SET NFS_SERVER /GID_DEFAULT=10
```

Changes the GID of the local server to 10.

SET ROUTE

Forces a specific routing path to a specified host or network.

Format

SET [NO]ROUTE ["destination"]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/[NO]CONFIRM	/CONFIRM
/DEFAULT_ROUTE	None.
/G_ADDRESS=internet_address	None.
/GATEWAY=gateway_spec	None.
/NETWORK	None.
/PERMANENT	None.

Restrictions

This operation requires OPER privilege if the Internet is active and read/write access to the UCX$ROUTE database.

Parameters

"destination"

Specifies the name or internet address of a host or a network as it is specified in the UCX$HOST or UCX$NETWORK database. The UCX$HOST database is checked first.

This parameter cannot be specified with the /ADDRESS or /DEFAULT qualifier.

Description

The SET ROUTE command forces a specific routing path to the specified host or network.

Logically there are two route databases, the permanent route database and the volatile route database. The permanent database resides on-disk. The volatile database resides in memory, when the network is active.

You can use the SET ROUTE command to affect either the permanent or the volatile database.

If the network is not active, the SET ROUTE command affects the permanent route database by default.

If the network is active, you must use the /PERMANENT qualifier to affect the permanent database. If the network is active and you do not specify the /PERMANENT qualifier, the changes affect the volatile database by default.

Command Qualifiers

/ADDRESS=internet_address
Specifies the destination Internet address of a host or network.

You must specify the /ADDRESS qualifier if you have not specified a destination using the *destination* parameter.

You cannot use the /ADDRESS qualifier in conjunction with the *destination* parameter.

It is recommended that you use the *destination* parameter instead of the /ADDRESS qualifier.

/[NO]CONFIRM
Disables or enables confirmation before route information is placed in the database during a wildcard delete operation. The default for wildcard delete operations is /CONFIRM.

/DEFAULT_ROUTE
Defines a default route. The default route defines the route on which a packet is sent if initial attempts to route the packet fail.

To define the default route, you must also specify the /GATEWAY qualifier, which defines the gateway for the default route.

You cannot use the /DEFAULT_ROUTE qualifier in conjunction with the *destination* parameter.

/G_ADDRESS=internet_address
Specifies the gateway Internet address of the gateway host.

You must specify the /G_ADDRESS qualifier if you have not specified the /GATEWAY qualifier. You cannot specify both the /GATEWAY qualifier and the /G_ADDRESS qualifier.

It is recommended that you use the /GATEWAY qualifier instead of this qualifier.

/GATEWAY=gateway_name
Specifies the gateway host name or address. You must first define the name or address with the SET HOST command. If you specified a host name, the name must appear in the host database.

You must specify the /GATEWAY qualifier if you have not specified the /G_ADDRESS qualifier. You cannot specify both the /GATEWAY qualifier and the /G_ADDRESS qualifier.

/NETWORK
Specifies that the route is a network route. Use the /NETWORK qualifier when a destination route is specified with the /ADDRESS qualifier and the route is a network route. If the /NETWORK qualifier is not specified, the destination is classified based on its Internet network class (A, B, or C). However, if the address is clearly a network number, the SET ROUTE command will interpret the number correctly.

Use the /NETWORK qualifier when the network number is ambiguous and could be interpreted as an Internet host address. For example, if there is a nonstandard network mask or the Internet address is abbreviated.

/PERMANENT
Specifies that the changes are to be made permanent to the UCX$ROUTE database only.

If the network is not active, the changes are made to the permanent route database by default.

If the network is active, you must use the /PERMANENT qualifier to affect the permanent database. If you do not specify the /PERMANENT qualifier, the changes affect the volatile database by default.

Examples

1
```
UCX> SET ROUTE DWHOAN/GATEWAY=GOLDEN_GATE
```

This command, issued from local host BRKLYN, provides a route so you can send a message from BRKLYN, which resides on network 100, to DWHOAN, which resides on network 101. It is possible to provide this specific route because on network 100, GOLDEN_GATE is a gateway to network 101.

The following list shows the host names and Internet addresses for the rest of the examples in this section:

- BRKLYN — address 100.45
- GOLDEN — address 100.42
- GOLDEN_GATE — address 101.42
- DWHOAN — address 101.81
- Default route — address 0.0

2
```
UCX> SET ROUTE 101.81/GATEWAY=100.42
```

This command is the equivalent of the preceding command using the Internet addresses, instead of the hosts' names.

3
```
UCX> SET ROUTE DWHOAN /GATEWAY=100.42
```

This command is the equivalent using both the host name and the Internet address.

4
```
UCX> SET ROUTE 100.45.0/GATEWAY=REMOTE/NETWORK
```

In this example, 100.45.0 looks like a host address and would be interpreted as such by the Connection. However, using the /NETWORK qualifier distinguishes it as a network address.

5
```
UCX> SET ROUTE/DEFAULT/GATEWAY=DEFGATE
```

Sets a default route with the host DEFGATE as the default gateway.

6
```
UCX> SET ROUTE/DEFAULT/GATEWAY=DEFGATE/PERMANENT
```

In this case, the information is added to the permanent on-disk database.

SHOW ARP

Displays ARP mapping information.

Format

SHOW ARP [host_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.

Restrictions

You cannot specify both a host name and a host Internet address.

Parameter

host_name
The name of a host that is a source or destination of ARP messages.

Each host name must have a unique Internet address.

Description

The SHOW ARP command displays ARP mapping information. If you do not provide a host name, all ARP information is displayed.

Command Qualifiers

/ADDRESS=internet_address
Specifies a host by an Internet address. You cannot specify both a host name and a host Internet address.

Example

```
UCX> SHOW ARP/ADDRESS=150.110.4.191
                                ARP table entries

Ethernet            Internet        Host name          ARP status
AA-00-04-00-67-FA   150.110.5.117   host_1             INUSE CMPL TRAIL
```

Displays information on the Ethernet, Internet, host name, and ARP status for Internet address 150.110.5.117.

SHOW BIND

Displays the characteristics of bound Connection file systems.

Format

SHOW BIND "filesystem_name"

Command Qualifiers	Defaults
None.	None.

Parameters

"filesystem_name"
The name given to the logical file system. This name is the first element of the ULTRIX FILE SPECIFICATION.

Description

The SHOW BIND command displays the characteristics of a bound Connection file system.

Example

```
UCX> SHOW BIND

Logical filesystem                    Pathname

$1$DUA8:                              /work2
$1$DUA7:                              /work1
UCX>
```

Lists all the bound file systems.

SHOW CFS

Monitors Connection file system statistics.

Format

SHOW CFS

Command Qualifiers	Defaults
/CONTINUOUS[=seconds]	/CONTINUOUS=4
/PARAMETERS	None.
/SUMMARY	None.

Description

The SHOW CFS command monitors Connection file system statistics.

Command Qualifiers

/CONTINUOUS[=seconds]
Selects the screen update interval in seconds. The default value is 4 seconds.

/PARAMETERS
Displays the active Connection file system environment parameters.

/SUMMARY
Displays CFS service wait states, data cache performance, and a summary of services and resources allocated.

Examples

The following command displays statistics on the file system.

```
UCX> SHOW CFS
    CFS Services                                VAX/VMS System Services
------------------------      ------------------------------------------------------
CLOSE                329
CREATE_FH             14      $ASSIGN            352      $QIO                     0
CREDIR_FH             22      $DASSGN            351          Access             315
FREEBUFF              06                                      Create              36
GETATTR              583      $DEQ             10020          Deaccess           351
LINK_FH                1      $ENQ             30806
LOOKUP_FH           4520                                      Read_attr         9580
OPEN_FH              315      $EXPREG            124          Write_attr         379
READ                 338      $SETPRT             58
READBUFF               0                                      Lookup            4774
READDIR_FH           140      $CLREF           6645
READLINK_FH            0      $SETEF           6645          Extend               9
REMDIR_FH              1
REMOVE_FH              7      $DCLAST          7536          Delete               8
RENAME_FH              1      $CLRAST            17          Enter                1
SETATTR              144      $SETAST         17135          Remove               1
STATFS                11
SYMLINK_FH             0      $GETDVI            15          Read_V             527
WRITE                219                                     Write_V            243
OTHER               1835      $CHKPRO          8678
TOTAL               8480
```

The following example displays CFS service wait states, data cache performance, and a summary of sevices and resources.

```
UCX> SHOW CFS /SUMMARY
                    CFS Service status and performance    27-SEP-1989 17:03:46.52
Service State     Cur  Max    Total     Cacheop  Cur  Hit  I/O  Inc     Status
---------------   ---  ---  ---------   -------  ----  ----  ----  ----  ------------
FP-Access           0    1         12   Read       0    0    0    0  Clusize    16
FP-Attributes       0    1         75   Read-A     0    0    0    0  Limit     256
FP-Delete           0    0          0   Write      0    0    0    0  Inuse       0
FP-Dir              0    1         17   Write-A    0    0    0    0  Busy        0
FP-Rename           0    0          0   Write-D    0                Hitrate     0
  FP Sub_total      0    1        104
Buffered            0    0          0   Nameop   Cur  Hit     Status
Internal            0    1          1   ------   ----  ----  -------------
I/O                 0    0          0   Add        0        Tabsize      0
Lock                0    0          0   Delete     0        Inuse        0
Logging             0    0          0   Lookup     0    0   Hitrate      0
Resource            0    0          0
RMS                 0    0          0   Fileop   Cur  Hit     Status
Service             0    1          1   ------   ----  ----  -------------
Timer               0    1         18   Find       0    0   Limit        0
Other               0    0          0   Find-A          0   Inuse        0
Synch               0    1         21   Find-C          0   Timeout      5
                                                           Hitrate      0

Services count      1    2         89   ATCBs:  2 TBABs: 32 RDCBs:   6 Pages:  2
```

The following example displays the statistics about the active Connection file system environment parameters.

```
UCX> SHOW CFS /PARAMETERS

                          CFS PARAMETERS

     Data cache
        Size                256
        Low limit           4
        High limit          8

     Attribute cache        Disabled
        Size                0
        Timeout             Not defined

     Name cache             Disabled
        Size                0

     Transfer size          8192
     Write size             8192

     Output fatal messages to device: _OPA0:
```

The following example displays the statistics on the file system and specifies that the screen be updated every 25 seconds.

```
UCX> SHOW CFS /CONTINUOUS=25
```

SHOW COMMUNICATION

Displays Internet parameters for the local host that was specified with the SET COMMUNICATION command.

Format

SHOW COMMUNICATION

Command Qualifiers	Defaults
/CONTINUOUS[=seconds]	/CONTINUOUS=4
/FULL	See text.
/MEMORY	See text.
/ROUTE	None.

Description

The SHOW COMMUNICATION command displays Internet parameters for the local host that was specified by the SET COMMUNICATION command.

Command Qualifiers

/CONTINUOUS[=seconds]
Selects the screen update interval in seconds. The default is 4 seconds.

/FULL
Provides detailed information for each buffer type. The /FULL qualifier is used only with the /MEMORY qualifier; it cannot be used with the /CONTINUOUS qualifier. The default is to give a brief listing of the information.

/MEMORY
Displays statistics for each buffer type. The default is to give a brief listing of the information.

/ROUTE
Displays routing statistics. The default is to give a brief listing of the information.

Example

```
UCX>  SHOW COMMUNICATION
Communication parameters

   Local host                          my_host
   Reassembly timer                         15
   Cluster timer                             1

                          Maximum      Current            Peak
   Device_sockets              50           15              32

                                 Minimum                          Maximum
   Buffers                  Number       Bytes           Number       Bytes
     Large                      10       17920              100      179200
     Small                      30        7680              330       84480

                             Free       Maximum
   IRPs                        20           200
                                     Socket buffer size quotas
   Protocol               Checksum      Receive            Send
     TCP                  enabled          4096            4096
     UDP                  enabled          9000            9000
     IP                   enabled

   IP forwarding          disabled      Broadcast    disabled
```

Displays the Internet parameters for the local host.

SHOW DEVICE_SOCKET

Displays socket information for the specified device-socket.

Format

SHOW DEVICE_SOCKET [device_socket_name]

Command Qualifiers	Defaults
/CONTINUOUS[=seconds]	/CONTINUOUS=4
/FULL	None.
/HOST=(host_name[,...])	None.
/PORT=[local_port]	None.
/SERVICE=service_list	None.
/TYPE=socket_type_list	None.

Parameters

device_socket_name
BG followed by the unit number of the device.

Description

The SHOW DEVICE_SOCKET command displays socket information for the specified device. The device-socket is composed of two parts: the Internet device and the socket. The SHOW DEVICE_SOCKET command uses the device name to identify the device-socket and obtains information about the socket.

Command Qualifiers

/CONTINUOUS[=seconds]
Specifies the screen update interval. This qualifier is valid only for a specific device-socket.

The default value is 4 seconds.

/FULL
Displays the following information for the specified device-socket:

- Local and remote ports

- Type
- Host
- Service
- Terminal
- Options
- State
- Receive buffer
- Send buffer
- All the counters associated with device-sockets

If you do not specify a device-socket, it displays the information for all the device-sockets.

/HOST=(host_name[,...])
Limits the display to the device-sockets that are bound to the specified remote hosts. You can use VMS wildcard characters to specify the host name.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

This does not include device-sockets bound to the ANY host (address 0.0.0.0).

/PORT=local_port
Limits the display to the device-sockets bound to the specified local port. If you do not specify a value, device-sockets for all the local ports are displayed.

/SERVICE=service_list
Limits the display to the device-sockets used for the specified local services. Supported services include the following:

- FTP
- NFS
- MOUNT

- PORTMAPPER
- rlogin
- Telnet

You can also specify /SERVICES=ALL to display device-sockets for all supported services.

TYPE=socket_type_list
Limits the display to the specified type of device-socket. Supported device-socket types include:

- DGRAM
- RAW_IP
- STREAM

You can also specify /TYPE=ALL to display device-sockets for all supported types.

Examples

1
```
UCX> SHOW DEVICE_SOCKET BG75 /CONTINUOUS =10
```
Displays information about device-socket 75 every 10 seconds.

2
```
UCX> SHOW DEVICE_SOCKET BG75 /FULL
```
Displays all the information about device-socket 75.

3
```
UCX> SHOW DEVICE_SOCKET /HOST=REMOTE
```
Displays information about all the device-sockets for the remote host REMOTE.

4
```
UCX> SHOW DEVICE_SOCKET /TYPE=(RLOGIN,TELNET)/HOST=REMOTE
```
Displays information about all the remote logins for the remote host REMOTE.

SHOW EXPORT

Displays which remote hosts are allowed access to a Connection file system.

Format

SHOW EXPORT "/ultrix_directory_spec"

Command Qualifiers	Defaults
/HOST=(host_name[,...])	None.
/OUTPUT=file_specification	See text.

Restrictions

This operation requires read access to the UCX$EXPORT database.

Parameters

"/ultrix_directory_spec"
Begins with a slash (/), followed by the file system name, and zero or more directory names separated by slashes.

Description

The SHOW EXPORT command displays which remote hosts are allowed access to a Connection file system. If you do not specify a file system, all exported file systems are displayed.

Command Qualifiers

/HOST=(host_name[,...])
Shows the exported file systems for the specified hosts.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

/OUTPUT=file_specification
Specifies that the output of the SHOW EXPORT command is output to the specified file. By default the output is displayed on the screen.

Example

```
UCX> SHOW EXPORT "/smith/work/jobs"
```

Displays which hosts have access to "/smith/work/jobs".

SHOW HOST

Displays information about a specified host.

Format

SHOW HOST [host_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/DOMAIN=domain_name	None.
/[NO]LOCAL	See text.
/OUTPUT=file_specification	See text.
/SERVER=(server_name[,...])	None.

Restrictions

This operation requires read access to the UCX$HOST database.

Parameter

host_name
The host name or Internet address of the host that is a source or destination of messages.

Each host name must have a unique Internet address.

You can use wildcards for the *host_name* parameter.

Description

The SHOW HOST command displays information from the local host database the specified host. If BIND is enabled, it also displays information on the specified host from the BIND database. If the *host_name* parameter is not specified, information about all the hosts is displayed.

All alias names for the specified host are displayed when you use the SHOW HOST command. If the specified host name is an alias, the host name and all alias names are displayed.

If the host has more than one Internet address and you specify the host name, then all Internet addresses and aliases for the host are displayed.

If the host has multiple Internet addresses and you specify an alias that is defined on multiple Internet addresses, then only the first Internet address and aliases for that host are displayed.

You can use wildcards for the *host_name* parameter.

Hosts in the BIND database are only displayed if you do not specify the *host_name* parameter, or you specify an asterisk. If you use an asterisk to complete a host name, no BIND information is displayed.

Qualifiers

/ADDRESS=internet_address
Allows you to select a host by Internet address.

If the host has more than one Internet address and you specify the host name, then all Internet addresses and aliases for the host are displayed.

If the host has multiple Internet addresses and you specify an alias that is defined on multiple Internet addresses, then only the first Internet address and aliases for that host are displayed.

It is recommended that you use the *host_name* parameter instead of this qualifier.

/DOMAIN=domain_name
Specifies the domain to be used by the local host. However, the definition of the domain name is only valid during the execution of the current SHOW HOST command. The BIND request is sent to the specified domain.

/[NO]LOCAL
Specifies whether the SHOW HOST command uses the local or the BIND database.

The /LOCAL qualifier specifies that only the local database is used.

The /NOLOCAL qualifier specifies that only the BIND database is used.

The default is to access both the local and BIND databases.

/OUTPUT=file_specification
Specifies that the output of the SHOW HOST command is output to the specified file. By default the output is displayed on the screen.

/SERVER=(server_name[,...])

Specifies the BIND servers to be used by the local host. However, the definition of the server name list is only valid during the execution of the current SHOW HOST command. The BIND request is sent to the specified server.

The server name can be either a host name or Internet address.

The host name list is ordered by request preference. For example, the initial request is sent to the first host in the list . If that host is unavailable, the request is sent to the second host in the list, and so on.

If you specify more than one server name, you must separate them with commas and enclose them in parentheses. Also, to ensure that each server name is explicit, enclose each in quotation marks. You can specify a maximum of three domains.

Examples

1 `UCX> SHOW HOST *`

Displays the entire UCX$HOST database.

In the following examples, the host has multiple Internet addresses and is defined as follows:

Internet Address	Host Name	Aliases
100.1	HOST_NAME	HOST_1A, HOST_1B, HOST_ALIAS
100.2	HOST_NAME	HOST_2A, HOST_2B, HOST_ALIAS

2 `UCX> SHOW HOST HOST_NAME`

Shows all the Internet addresses and aliases for the host HOST_NAME.

3 `UCX> SHOW HOST HOST_1A`

Shows Internet address 100.1 and the aliases HOST_1A, HOST_1B, and HOST_ALIAS.

4 `UCX> SHOW HOST HOST_ALIAS`

Shows the host and all aliases for addresses 100.1 and 100.2.

SHOW INTERFACE

Displays information about a specific communication interface.

Format

SHOW INTERFACE [interface_name]

Command Qualifiers	Defaults
/CLUSTER	None.
/CONTINUOUS[=seconds]	/CONTINUOUS=4

Parameter

interface_name
A 2-character string followed by a unit number or a wildcard character (*).

Description

The SHOW INTERFACE command displays information about a specific interface. If you do not specify an interface name, information about all the interfaces on the host is displayed.

Command Qualifiers

/CLUSTER
Displays information about the cluster of which the interface is a member.

/CONTINUOUS[=seconds]
Specifies the screen update interval in seconds. The default value is 4 seconds. You must specify a specific interface to use this qualifier.

Example

```
UCX> SHOW INTERFACE lo0
Interface: lo0
   IP_Addr: 127.0.0.1          NETWRK: 255.0.0.0        BRDCST:
                        Ethernet_Addr:
     Flags: UP LOOP
                                  RECEIVE          SEND
   Packets                              6           274
     IP packets                         0           274
     Broadcast IP packets               0             0
     Trailer 1 IP packets               0
     Trailer 2 IP packets               0
     ARP packets                        0             0
     Broadcast ARP packets              6             0
   Drops
     IP packets                         0             0
     ARP packets                        0             0
   Errors
     Hardware                           0             0
     Software                           0             0
   Restarting attempts                  0
   Successful restarts                  0
```

Displays information about interface device de0.

SHOW NAME_SERVICE

Displays all information pertaining to the name service.

Format

SHOW NAME_SERVICE

Command Qualifiers	Defaults
None.	None.

Description

The SHOW NAME_SERVICE command displays all information pertaining to the name service. The display includes both system and process-specific parameters.

Examples

1

```
UCX> SHOW NAME_SERVICE

BIND Resolver Parameters

 System

  State:     Enabled
  Transport: UDP
  Domain:    abc.xyz.com
  Servers:   130.180.34.2, 130.180.15.34, 130.180.6.160

 Process

  Transport:
  Domain:
  Servers:   130.180.34.3
```

In this example, the process-specific server has been set to 130.180.34.3; therefore, any commands that access BIND issued from this process will use the specified server. Commands issued from other processes will access the system servers listed.

SHOW NETWORK

Displays information about a specified network.

Format

SHOW NETWORK [network_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/OUTPUT=file_specification	See text.

Restrictions

This operation requires read access to the UCX$NETWORK database.

Parameter

network_name
An ASCII string that identifies a network number.

Description

The SHOW NETWORK command displays information about a specified network. If you do not specify a network name, information about all the networks on the Ethernet is displayed.

All alias names of the specified network are displayed. If the specified network is an alias, the network name and all the alias names are displayed.

Command Qualifiers

/ADDRESS=internet_address
Selects a network by its Internet address. This qualifier cannot be used if a network name has been specified.

/OUTPUT=file_specification
Specifies that the output of the SHOW NETWORK command is output to the specified file. By default the output is displayed on the screen.

Examples

1

```
UCX>   SHOW NETWORK NYNET
Network address     Network name

4.0.0.0             NYNET
```

Displays information about the network named NYNET.

2

```
UCX>   SHOW NETWORK Z*
Network address     Network name

138.180.4.0         zznet, ZZNET
120.45.30.0         zzo-net, ZZO-NET, zz01-net
```

Displays information about all networks with names or aliases beginning with the letter Z.

SHOW NFS_SERVER

Displays NFS server information and monitors the performance of the NFS server.

Format

SHOW NFS_SERVER

Command Qualifiers	Defaults
/CONTINUOUS[=seconds]	/CONTINUOUS=4
/HOST=host_name	None.
/IDENTIFICATION	None.
/PARAMETERS	None.
/SERVICES	None.
/USER_NAME=vms_username	None.

Restrictions

This operation requires SYSNAM and WORLD privileges.

Description

Displays NFS server information and monitors the performance of the NFS server.

Command Qualifiers

/CONTINUOUS[=seconds]
Specifies the screen update interval in seconds; the default is 4 seconds. You must specify a specific interface to use this qualifier. You cannot use this qualifier with the /PARAMETERS qualifier.

/HOST=host_name
Used in combination with the /USER_NAME qualifier to select a client from the UCX$PROXY database.

/IDENTIFICATION
Provides static information about the NFS server.

/PARAMETERS
Displays NFS parameters that were set with the SET NFS_SERVER command.

/SERVICES
Displays statistics for each of the services provided by NFS: Mount, NFS, and PORTMAPPER.

/USER_NAME=vms_username
Used in combination with the /HOST qualifier to display information about the services used by the specified VMS user name. The /USER_NAME qualifier does not accept a list of user names (or account names).

Example

```
UCX> SHOW NFS_SERVER
```

Displays information about the local server.

See Chapter 4 for more information on NFS.

SHOW PROTOCOL

Displays protocol counters.

Format

SHOW PROTOCOL [protocol_type]

Command Qualifiers	Defaults
/CONTINUOUS[=seconds]	/CONTINUOUS=4

Parameter

protocol_type
One of the following protocols can be specified: IP, TCP, UDP, or ICMP. The default is to display information on all the protocols.

Description

The SHOW PROTOCOL command displays the protocol counters for the specified protocol. If you do not specify a protocol, counters for all the protocols are displayed.

Command Qualifiers

/CONTINUOUS[=seconds]
Specifies the screen update interval in seconds; the default value is 4 seconds. You must specify a specific protocol to use this qualifier.

Example

```
UCX> SHOW PROTOCOL/CONTINUOUS=10 IP
 IP
   Bad IP Version drops:                                    0
   Unknown IP broadcast address drops:                      0
   Datagrams dropped:                                       0
   Total datagrams:                                   1050283
   Datagrams with bad header checksums:                     0
   Datagrams with size smaller than minimum:                0
   Datagrams with data size < data length:                  0
   Datagrams with header length < data size:                0
   Datagrams with data length < header length:              0
   Datagrams forwarded:                                     0
   Fragments received:                                   1163
   Fragments dropped:                                       0
   Fragments timed out:                                     0
   Packets received for unreachable destination:       100498
   Packets forwarded on same net:                           0
   Fragments dropped from reassembly queue:                 0
   Fragments queued in reassembly queue:                 1163
   Datagrams dropped during reassembly:                     0
   Datagrams successfully reassembled:                    229
```

Displays the counter values for the IP protocol and updates the display every 10 seconds.

SHOW PROXY

Displays information about the VMS accounts that are used by the ULTRIX clients.

Format

SHOW PROXY [vms_account]

Command Qualifiers	Defaults
/GID=gid	/GID=*
/HOST=(host_name[,...])	/HOST=*
/OUTPUT	See text.
/PERMANENT	See text.
/UID=uid	/UID=*

Restrictions

This operation requires read access to the UCX$PROXY database.

Parameter

vms_account
An account name found in the VMS system user authorization file (SYSUAF.DAT).

Description

The SHOW PROXY command displays information about the VMS accounts that are used by ULTRIX clients using the Connection software.

The *vms_account* parameter specifies an account name found in the VMS system user authorization file.

If you do not specify an account with the *vms_account* parameter, information is displayed for all proxy VMS accounts.

Command Qualifiers

/GID=gid
Specifies the ULTRIX GID of the client for which information is to be displayed. If you do not specify this qualifier, information is given for all clients.

/HOST=(host_name[,...])
Specifies the ULTRIX hosts from which information is to be displayed. If you do not specify this qualifier, information is given for all hosts.

If you specify more than one host, you must separate them with commas and enclose them in parentheses. Also, to ensure that each host name is explicit, enclose each in quotation marks.

/OUTPUT=file_specification
Specifies that the output of the SHOW PROXY command is output to the specified file. By default the output is displayed on the screen.

/PERMANENT
Specifies whether or not NFS is queried before displaying each entry of the permanent, on-disk database.

If NFS is active and you specify /PERMANNENT, the volatile database is not queried.

If NFS is active and you do not specify the /PERMANENT qualifier, the volatile database is queried for each entry to see if it is known. Known entries are indicated by an asterisk (*) in the far righthand column.

/UID=uid
Specifies the ULTRIX UID of the client for which information is to be displayed. If you do not specify this qualifier, information is given for all clients.

Example

```
UCX>  SHOW PROXY
VMS User_name                    User_ID     Group_ID  Host name

BANDITT                              330           10  *                          *
BEEFEVEN                             457           10  *                          *
%UCX-E-NOUSER, No such user
CARDOMINO                            424           10  *
    .
    .
    .
wardwood                             386           10  *                          *
yelgen                               387           18  *                          *
%UCX-E-NOUSER, No such user
zappy                                450           10  *
zimmell                             1062           10  *                          *
```

Shows all VMS accounts for all clients. Note that the volatile database entries are indicated by asterisks (*) the far righthand column.

```
UCX>  SHOW PROXY /HOST=GOLDEN
```

Shows all VMS accounts for which any client from host GOLDEN has authorization.

SHOW ROUTE

Displays the permanent and volatile routing databases.

Format

SHOW ROUTE [destination_name]

Command Qualifiers	Defaults
/ADDRESS=internet_address	None.
/FULL	See text.
/G_ADDRESS=internet_address	None.
/GATEWAY=gateway_name	None.
/OUTPUT=file_specification	See text.
/PERMANENT	See text.

Restrictions

You cannot specify the name and address for the same entity.

This operation requires read access to the UCX$ROUTE database.

Parameter

destination_name
Host name of the destination host.

Description

The SHOW ROUTE command displays the permanent and volatile routing databases. The database that is displayed depends on whether the Internet is active and whether you specify the /PERMANENT qualifier.

The specified destination name is first looked up in the UCX$HOST database. If this lookup fails, the destination name is looked up in the UCX$NETWORK database.

Command Qualifiers

/ADDRESS=internet_address
Specifies the destination Internet address, which can be the name of a host or network. The Internet software determines whether the specified address is for a host or a network.

It is recommended that you use the *destination_name* parameter instead of this qualifier.

/FULL
Specifies that a mapping between the destination addresses and names and gateway addresses and names is included in the display.

If you specify the SHOW ROUTE command without /FULL, the default is to display the route as specified in the SET ROUTE command.

/G_ADDRESS=internet_address
Specifies the gateway Internet address of the gateway host.

You must specify the /G_ADDRESS qualifier if you have not specified the /GATEWAY qualifier. You cannot specify both the /GATEWAY qualifier and the /G_ADDRESS qualifier.

It is recommended that you use the /GATEWAY qualifier instead of the /G_ADDRESS qualifier.

/GATEWAY=gateway_name
Specifies the gateway host name. This name must first have been defined with the SET HOST command.

You must specify the /GATEWAY qualifier if you have not specified the /G_ADDRESS qualifier. You cannot specify both the /GATEWAY qualifier and the /G_ADDRESS qualifier.

/OUTPUT=file_specification
Specifies that the output of the SHOW ROUTE command is output to the specified file. By default, the output is displayed on the screen.

/PERMANENT
Specifies that only the permanent UCX$ROUTE database information will be displayed.

If the Internet is active and you do not specify the /PERMANENT qualifier, the volatile database is displayed by default.

If the Internet is not active the permanent database is displayed by default.

Examples

1

```
UCX>  SHOW ROUTE/FULL
Type            Destination                          Gateway

N    * 150.111.0.0                          150.110.5.118    gate_host
H      150.111.4.10   destin_host           150.110.5.120    gate_host_2
UCX>
```

Displays the network route information.

Route types include the following:

- N — Network route
- H — Host route
- A — Automatic route (generated by the UCX command SET INTERFACE)

An asterisk (*) next to the destination Internet address means that it was specified with the /ADDRESS qualifier.

An asterisk next to the gateway Internet address means that it was specified with the /G_ADDRESS qualifier.

If there is an entry in the UCX$HOST or UCX$NETWORK database for a route that was entered by address, a SHOW ROUTE command does not display the route, if the name for that address was specified. You display the route either by specifying its address or by specifying a wildcard.

If there is no host name next to the internet address when you use the /FULL qualifier, it means that there is no entry in the UCX$HOST or UCX$NETWORK database for this address.

If you use the /FULL qualifier and specify a route by name and the entry in the UCX$HOST or UCX$NETWORK database is deleted, then an error message is displayed.

2

```
UCX>  SHOW ROUTE "rabbit"
```

Displays the network route to host "rabbit".

START COMMUNICATION

Activates the Internet communications.

Format

START COMMUNICATION

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires the operator (OPER) privilege.

Description

The START COMMUNICATION command starts the Internet software. The buffer pool is created, and the IP, TCP, UDP, and ICMP protocols are started.

You must issue this command before you activate any of the Internet interfaces. When you issue this command, a message is sent to the network operator's console.

If you issue this command to start the Internet you will also have to issue several other commands to full start the Internet. Therefore, it is recommend that instead of using this command that you use the UCX$INET_STARTUP.COM command procedure (located in SYS$MANAGER) to start the Internet. This command procedure contains all the commands required to start the Internet.

Example

```
UCX> START COMMUNICATION
```

Activates the Internet communications.

START SERVICE

Initializes communication for the specified service or services.

Format

START SERVICE [service_name]

Command Qualifiers	Defaults
/ALL	None.

Parameter

service_name
Name of the service you want to start.

You must specify a specific service, unless you use the /ALL qualifier.

Description

The START SERVICE command initializes communication for the specified service or services. Supported services include Telnet and rlogin.

You can issue this command only after the Internet software has been initialized (see START COMMUNICATION), and the Telnet and rlogin server has been loaded.

The /ALL qualifier is required to start all known services.

You cannot specify the /ALL qualifier with a service name.

When you use this command, a message is sent to the log file and displayed on the network operator's console.

Command Qualifier

/ALL
Specifies that all supported services should be started.

Examples

1 `UCX> START SERVICE TELNET`

Activates Telnet communications.

2 `UCX> START SERVICE/ALL`

Starts all known services.

START ROUTING

Starts the dynamic routing server.

Format

START ROUTING

Command Qualifiers	Defaults
/SUPPLY=(option)	None.
/[NO]LOG	None.

Description

The START ROUTING command starts the dynamic routing server, which enables the host to advertise its routing information to other hosts and networks it is connected to.

When you use this command, a message is sent to the log file and displayed on the network operator's console.

Command Qualifiers

/SUPPLY=(option)
Specifies that the host will supply its routing information to other hosts on the network. Routing information is supplied every 30 seconds.

The /SUPPLY=DEFAULT option specifies that the specified host will supply the default network route; in which case, all packets that cannot be routed to their destination host or network are routed through the default host.

/[NO]LOG
Specifies whether routing activity is output to a log file.

Examples

1
```
UCX> START ROUTING
```

Starts the dynamic routing program. In this case, this host will receive dynamic routing information from other hosts on the network, but will not supply routing information to other hosts.

2 `UCX> START ROUTING/SUPPLY`

Starts the dynamic routing program. The /SUPPLY qualifier indicates that this host will broadcast its routing information to other hosts on the network in addition to receiving routing information.

STOP COMMUNICATION

Shuts down the Internet communications.

Format

STOP COMMUNICATION

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires the operator (OPER) privilege.

Description

The STOP COMMUNICATION command halts the execution of the Internet software and all services (Telnet and rlogin) are terminated.

All I/O operations are terminated. All the device-sockets are closed, pending cancellation of I/O operations. All buffers are deallocated, all interfaces are deleted, and all automatic routes are deleted from the UCX$ROUTE database.

The Internet software must be reloaded and restarted to continue Internet communications.

When you use this command, a message is sent to the log file and displayed on the network operator's console.

If you use this command to shut down the Internet, there are several other commands you must issue. Therefore, it is recommended that you use the UCX$INET_SHUTDOWN.COM command procedure (located in SYS$MANAGER) to shut down the Internet. This command procedure contains all the commands required to shut down the Internet in an orderly fashion.

Example

```
UCX> STOP COMMUNICATION
```

Shuts down the Internet communications.

STOP ROUTING

Stops the dynamic routing server.

Format

STOP ROUTING

Description

The STOP ROUTING command stops the dynamic routing server; therefore the host does not advertise its routing information to other hosts and networks it is connected to.

When you use this command, a message is sent to the log file and displayed on the network operator's console.

Example

```
UCX> STOP ROUTING
```

Stops the supply of routing information to the network.

STOP SERVICE

Stops the specified service or services.

Format

STOP SERVICE [service_name]

Command Qualifiers	Defaults
/ALL	None.

Parameter

service_name
Name of the service you want to stop.

You must specify a specific service, unless you use the /ALL qualifier.

Description

The STOP SERVICE command stops the specified service or services. All active communication for the specified services are aborted. Supported services include Telnet and rlogin.

You can issue this command only after the Internet software has been initialized (see START COMMUNICATION).

The /ALL qualifier is required to stop all known services.

You cannot specify the /ALL qualifier with a service name.

When you use this command, a message is sent to the log file and displayed on the network operator's console.

Command Qualifier

/ALL
Causes all supported services to be stopped.

Examples

1 `UCX> STOP SERVICE TELNET`

Deactivates Telnet communications.

2 `UCX> STOP SERVICE /ALL`

Stops all known services.

UNBIND

Undefines a Connection file system that was previously bound.

Format

UNBIND "filesystem_name"

Command Qualifiers

None.

Defaults

None.

Restrictions

This operation requires the SYSPRV privilege and may require the BYPASS privilege.

Parameter

"filesystem_name"

The name given to the logical file system. This name is the first element of the ULTRIX file specification.

Description

The UNBIND command undefines a Connection file system that was previously bound.

The UNBIND command is the logical equivalent of the VMS DISMOUNT command, in that the VMS NFS server can no longer access the file system.

Example

```
UCX> UNBIND "/disk@host"
```

Undefines the file system "/disk@host" from being bound.

ZERO INTERFACE

Clears statistics for the specified Internet interface.

Format

ZERO INTERFACE interface_name

Command Qualifiers

None.

Defaults

None.

Restrictions

This operation requires the operator (OPER) privilege.

Parameter

interface_name

A 2-character string followed by a unit number. The unit number is 0 for the first Ethernet controller, 1 for the second Ethernet controller, and so on. The following table lists the interface names and their corresponding Ethernet controllers.

Interface Names	Description
lo0	The local interface (loopback)
se0, se1,...	ESA, ESB,... (DESVA Ethernet controller)
qe0, qe1,...	XQA, XQB,... (DEQNA, DELQA, DEQTA Ethernet controller)
de0, de1,...	XEA, XEB,... (DEUNA, DELUA Ethernet controller)
ni0, ni1,...	ETA, ETB,... (DEBNA, DEBNE Ethernet controller)
ze0, ze1,...	EZA, EZB,... (SEGEC Ethernet controller)
xe0, xe1,...	EXA, EXB,... (DEMNA Ethernet controller)

Description

The ZERO INTERFACE command clears the statistics for the specified Internet interface.

If you do not specify an Internet interface name, all counters for all the interfaces are cleared.

Example

```
UCX> ZERO INTERFACE qe0
```

Clears the statistics for the qe0 interface.

ZERO NFS_SERVER

Clears the NFS server statistics.

Format

ZERO NFS_SERVER

Command Qualifiers	Defaults
/HOST=host_name	None.
/SERVICES	None.
/USER_NAME=[vms_username]	None.

Restrictions

This operation requires SYSNAM and WORLD privileges.

Description

The ZERO NFS_SERVER command clears the statistics of the NFS server.

Command Qualifiers

/HOST=host_name
Used in combination with the /USER_NAME qualifier to clear the statistics kept for that user name on specified hosts.

/SERVICES
Used to zero statistics for the named service. If you do not specify a name, statistics for all NFS services are cleared.

/USER_NAME=[vms_username]
Used in combination with the /HOST qualifier to clear statistics concerning a particular user or users that share the named VMS account. The /USER_NAME qualifier does not accept a list of account names.

Example

```
UCX>  ZERO NFS_SERVER /USER_NAME=WORK /HOST="june"
```

Clears the NFS server statistics for host "june" in using the VMS account WORK.

ZERO PROTOCOL

Clears the protocol counters.

Format

ZERO PROTOCOL [protocol_name]

Command Qualifiers	Defaults
None.	None.

Restrictions

This operation requires the operator (OPER) privilege.

Parameter

protocol_name
Can be one of the following protocols: IP, TCP, UDP, or ICMP.

Description

Clears the protocol counters kept by the Internet software. If you do not use the *protocol_name* parameter to specify a specific protocol, all protocol counters are cleared.

Example

```
UCX> ZERO PROTOCOL UDP
```

Clears the UDP protocol counters.

A

Connection Error Messages

A.1 Display Terminal Errors Messages

This section provides information on Connection errors that are directed to the display terminal.

%UCX-I-BIND_NONAUTH, The server was changed, but it is non-authoritative

Explanation: The address for the given server was not in the BIND database. A non-authoritative server is a slave server that does not have control over the data. The server gets information from another server (primary or secondary), caches the information, and updates the information periodicallly. Therefore, the answer provided by this server may not be correct.

User Action: Check that the given server name has been defined for the system.

%UCX-E-BIND_NOSERVERS, Default servers are not available

Explanation: There are no BIND servers defined.

User Action: Run SYS$MANAGER:UCX$CONFIG to define the default BIND servers.

%UCX-W-BIND_NOSERVNAM, Cannot find server name for address

Explanation: The server name for the given address is not in the local database.

User Action: Check that the given server address has been defined for the system.

%UCX-E-BIND_READERR, Error reading/interpreting query response

Explanation: The BIND query to the remote server was not completed due to Internet communication problems.

User Action: Check that your Internet is properly configured and running, and try again. You may also check the communications with the remote servers by using the UCX LOOP command.

%UCX$-E-DUPHOSTNAME, Duplicate UCX$HOST host name for <host_name>

Explanation: An additional Internet address has been defined for a host that already exists in the database.

User Action: None.

%UCX$-I-HOSTADD, Host information added to database

Explanation: This message is displayed along with the UCX$-I-DUPHOSTNAME message.

User Action: None.

%UCX$-F-NFS_ABORT, NFS server XXXXXX aborted

Explanation: Caused by an internal NFS fatal error that is preceded by an NFS error.

User Action: Examine the log file.

%UCX$-I-NFS_ACCDEF, Failed to delete account XXXX from volatile database

Explanation: The record was not found in the volatile database.

User Action: Check the record information supplied with the REMOVE PROXY command.

%UCX$-I-NFS_ACCDES, Deleted account XXXX from volatile database

Explanation: The record was deleted from the volatile database.

User Action: None.

%UCX$-W-NFS_ACCNOA, Access to the VMS account XXXX is denied

Explanation: Someone attempted to access the NFS server over the network when network access was restricted on account XXXX.

User Action: See your system manager to determine at what times network access is allowed. You must have network access.

%UCX$-F-NFS_ALLMEM, Failed to allocate virtual memory in module XXXXXX

Explanation: Memory is needed to open another file, but the memory quota was exceeded for the NFS process.

User Action: Change the NFS startup file (virtual memory limit or page file quotas).

%UCX$-F-NFS_AUTHER, Invalid RPC authentication

Explanation: Did not find entry in proxy database.

User Action: Check for possible intruder.

%UCX$-F-NFS_BCPCHH, Failed to assign channel to mailbox XXXXXX

Explanation: A VMS system call failed.

User Action: See the error code returned. Check process quotas, privileges, and available memory.

%UCX$-F-NFS_BCPMBR, Failed to read from mailbox

Explanation: A VMS system call failed.

User Action: Examine the error code. Check the process quotas, privileges, and available memory.

%UCX$-F-NFS_BCPMBW, Failed to write into mailbox

Explanation: A VMS system call failed.

User Action: Examine the error code returned. Check process quotas, privileges, and available memory.

%UCX$-F-NFS_BCPWAA, Failed to set write attention for mailbox

Explanation: There is a VMS call that has failed.

User Action: Examine the error code returned. Check process quotas, privileges, and available memory.

%UCX$-E-NFS_BFSCAL, XXXXXX call failure

Explanation: Informational error returned by the Connection file system. The NFS server provides mapping between VMS/RMS errors and NFS errors.

User Action: Depends on the VMS/RMS error code, returned by the NFS process. To analyze the situation, supply the system manager with the expanded VMS/RMS error codes.

%UCX$-E-NFS_BFSCLS, CLOSE call failure

Explanation: There is an error in closing a file. The error could be caused by a file protection violation or if the file was being accessed when an attempt was made to close the file.

User Action: See the error code returned.

%UCX$-F-NFS_DEAMEM, Failed to deallocate virtual memory in module XXXXXX

Explanation: Internal NFS error deallocating memory.

User Action: Submit an SPR and provide the following:

- Attempt to isolate the cause of the problem to one client and the sequence of the steps.
- Describe the configuration of the client and server hosts.
- Submit error and output files to Digital.

%UCX$-F-NFS_DEFLNM, Failed to create logical name XXXXXX

Explanation: Failure to create one of the run-time logical names.

User Action: Check default logical names that are set by the NFS startup command file for being present and its values.

%UCX$-I-NFS_DISERR, Disable writing into the NFS error log file

Explanation: A SET NFS_SERVER/DISABLE=reporting_option command was issued.

User Action: None.

%UCX$-F-NFS_DUPUID, Duplicate NFS proxy information for host XXXXXX

Explanation: Information for the specified host was already found in the proxy database. Second occurrence is ignored.

User Action: Correct the proxy database.

%UCX$-I-NFS_ENAERR, Enable writing into the NFS error log file

Explanation: A SET NFS_SERVER/ENABLE=reporting_option command was issued.

User Action: None.

%UCX$-I-NFS_ERRLOG, Failed to write into log file

Explanation: A VMS error will follow this error. The log file becomes disabled and will remain disabled until the SET NFS_SERVER command is issued.

User Action: Examine the error message and take appropriate action.

%UCX$-F-NFS_EXPBCP, Failed to access EXPORT database

Explanation: There was an error in accessing the export database.

User Action: Depends on the VMS/RMS error code returned by the NFS process. Possible causes include deleting export file or changing protection.

%UCX$-W-NFS_EXPNRC, Failed to find in EXPORT database file system XXX

Explanation: Unauthorized attempt to mount a file system. It could be an intruder attempt.

User Action: Check for possible intruder.

%UCX$-F-NFS_FATERR, NFS server internal error in module XXXXXX

Explanation: Internal program error in the NFS code.

User Action: Submit an SPR and:

- Attempt to isolate the cause of the problem to one client and the sequence of the steps.
- Describe the configuration of the client and server hosts.
- Submit error and output files to Digital.

%UCX$-F-NFS_FOPLIM, Exceeded FILLM quota or process channel limit

Explanation: Open files quota for the particular user is exceeded.

User Action: May need to modify FILLM quota in the account with the AUTHORIZE utility.

%UCX$-S-NFS_HSTNAM, NFS server is running on host XXXXXX

Explanation: Informational.

User Action: None.

%UCX$-F-NFS_INECTL, Error translating host name to host address

Explanation: Internet software error translating host name into host address.

User Action: Enter the server host name into the host database.

%UCX$-F-NFS_INIERR, Failed to start NFS server initialization

Explanation: Fatal error during initialization before log file was created.

User Action: Examine the additional NFS or VMS error messages on the operator console.

%UCX$-F-NFS_INEGBN, Failed to find host XXXXXX in host database

Explanation: Specified host is not found in host database.

User Action: Add host name and address to the host database.

%UCX$-F-NFS_INETAS, Failed to assign channel to network device

Explanation: Failure to assign new channel to the device.

User Action: Check quotas of channels that are available to the process.

UCX$-E-NFS_INETWH, Error writing to host XXXX (no name) on socket YYY

Explanation: During a read or write operation, the NFS server attempted to send a message to an unreachable host. NFS could not translate the remote host name into an address.

User Action: None.

%UCX$-F-NFS_INETRE, Error reading from socket

Explanation: Error reading from the network. A VMS error message will follow this message.

User Action: Examine the VMS status code and determine the cause of the error.

%UCX$-F-NFS_INETST, Error creating or setting socket parameters

Explanation: A VMS system call failed.

User Action: Verify if the NFS process privileges that are set in the NFS startup procedure are correct, then take the appropriate action.

%UCX$-E-NFS_INETWR, Error writing to socket

Explanation: Error writing to the network. A VMS error message will follow this error message.

User Action: Examine the returned error code.

%UCX$-F-NFS_INICMB, Failed to create mailbox XXXXXX

Explanation: NFS process failed to create a communication channel with the Connection Management module.

User Action: Correct the problem specified by the VMS error code. Possible causes are byte limits, quota, privilege violation (requires SYSNAM privilege), and insufficient memory.

%UCX$-F-NFS_INIEXP, Failed to initialize EXPORT database

Explanation: There is an error during the initialization stage of accessing the export data base. See the returned status code.

User Action: Verify the export file accessibility (presence, device, protection, and internal RMS errors).

%UCX$-F-NFS_INIGPI, Failed to create system logical name XXXXXX

Explanation: The VMS system call failed to create the logical name. A VMS error message will follow this message.

User Action: Examine the error message and take appropriate action.

%UCX$-F-NFS_INIINE, Failed to establish network communication

Explanation: Failed to initialize the Internet communication software.

User Action: Examine the error message and take appropriate action.

%UCX$-F-NFS_INILNM, Failed to find server host logical name

Explanation: None of the server host logical names (UCX$INET_HOST or UCX$NFS_HOST) are defined. If UCX$INET_HOST is not defined, then it is likely that there is no active Internet interface.

User Action: Check all default logical names that are set by the NFS startup process for being present and their values. Correct the error.

%UCX$-F-NFS_INIMAN, Failed to initialize NFS management subsystem

Explanation: Failure to initialize connection with the NFS management module.

User Action: Examine the error message and take appropriate action.

%UCX$-F-NFS_INIMEM, Failed to initialize virtual memory in module XXXXXX

Explanation: The cause of the error could be one of the following:

- Virtual memory allocation failure caused within the NFS process or a page file quota violation.
- There is a proxy or export database initialization problem. The problem is caused by errors during the building of the internal database structure.

User Action: Change NFS process parameters.

%UCX$-F-NFS_INIPDB, Failed to access proxy database

Explanation: The NFS process cannot access the proxy database.

User Action: Verify the proxy database file accessibility.

%UCX$-F-NFS_INIRDB, Failed to get rights information for user XXXXXX

Explanation: The VMS Rights database does not include the expected rights identifier for user XXXXXX.

User Action: Verify the Rights database information for account XXXXXX.

%UCX$-F-NFS_INISCR, UCX$NFS_REMOTE identifier missing from Rights database

Explanation: NFS_SECURITY logical is set to provide security features, but UCX$NFS_REMOTE identifier is not added to the Rights database. This identifier is normally added to the Rights database when the Connection is installed.

User Action: Add the UCX$NFS_REMOTE identifier to the Rights database.

%UCX$-F-NFS_INISPI, Invalid NFS server parameter(s) detected

Explanation: One of the parameters of the NFS server that are defined by the logical names has an unappropriate value.

User Action: Check default logical names that are set by the NFS startup command file and their values.

%UCX$-F-NFS_INITHR, Failed to initialize multiple threads

Explanation: An error occurred during the initialization of the multithreading environment. There is a possible virtual memory allocation failure or page file quota violation. See the returned VMS status.

User Action: Check the default logical values UCX$NFS_THREAD and UCX$NFS_XID to be sure these values are valid and the NFS process parameters in the NFS startup file are correct.

%UCX$-F-NFS_INIUAI, Failed to get account information for user XXXXXX

Explanation: Any proxy database entry using XXXXXX will be disabled because this account was not found.

User Action: Verify that account XXXXXX is in the VMS SYSUAF file.

%UCX$-F-NFS_INIVDB, Failed to initialize proxy database

Explanation: Failed to initialize the proxy database. A VMS error message will follow this error message.

User Action: Examine the error message and take appropriate action.

%UCX$-S-NFS_INISUC, NFS server is starting using log file XXXXXX

Explanation: Informational.

User Action: None.

%UCX$-W-NFS_INTRUD, Inconsistent host database or intruder attempt

Explanation: The host Internet address cannot be verified. It could be a possible intruder.

User Action: Check for possible intruder.

%UCX$-I-NFS_LGFCLO, Failed to close log file

Explanation: A VMS error will follow this error. The log file becomes disabled and will remain disabled until the SET NFS_SERVER command is issued.

User Action: Examine the error message and take appropriate action.

%UCX$-I-NFS_LGFCRE, Failed to create log file

Explanation: A VMS error will follow this error. The log file becomes disabled and will remain disabled until the SET NFS_SERVER command is issued.

User Action: Examine the error message and take appropriate action.

%UCX$-F-NFS_LNMDBG, Failed to translate logical name (UCX$NFS00000000_DEBUG)

Explanation: The logical name UCX$NFS00000000_DEBUG could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMERR, Failed to translate logical name (UCX$NFS00000000_ERROR)

Explanation: The logical name UCX$NFS00000000_ERROR could not be translated.

User Action: Check value of the logical name in the startup command file.

%UCX$-F-NFS_LNMGID, Failed to translate logical name (UCX$NFS00000000_GID)

Explanation: The logical name UCX$NFS00000000_GID could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMHST, Failed to translate logical name (UCX$NFS00000000_HOST)

Explanation: The logical name UCX$NFS00000000_HOST could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMINA, Failed to translate logical name (UCX$NFS00000000_INACTIVITY)

Explanation: The logical name UCX$NFS00000000_INACTIVITY could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMOPC, Failed to translate logical name (UCX$NFS00000000_OPCOM)

Explanation: The logical name UCX$NFS00000000_OPCOM could not be translated.

User Action: Check value of the logical name in the startup command file.

%UCX$-F-NFS_LNMSEC, Failed to translate logical name (UCX$NFS00000000_SECURITY)

Explanation: The logical name UCX$NFS00000000_SECURITY could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMTDF, Failed to translate logical name (UCX$NFS_TIME_DIFFERENTIAL)

Explanation: The logical name UCX$NFS00000000_DIFFERENTIAL could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMTHR, Failed to translate logical name (UCX$NFS00000000_THREADS)

Explanation: The logical name UCX$NFS00000000_THREADS could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMUID, Failed to translate logical name (UCX$NFS00000000_UID)

Explanation: The logical name UCX$NFS00000000_UID could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-F-NFS_LNMXID, Failed to translate logical name (UCX$NFS00000000_XID)

Explanation: The logical name UCX$NFS00000000_XID could not be translated.

User Action: Check the value of the logical name in the startup command file.

%UCX$-I-NFS_MNTSUC, Mounted file system XXXXXX

Explanation: Informational.

User Action: None.

%UCX$-I-NFS_NOACNT, No NFS clients authorized to use server

Explanation: No records in proxy database with valid VMS accounts.

User Action: Check proxy database and VMS accounts in AUTHORIZATION file.

%UCX$-F-NFS_NOAMAP, Attempted to set file ownership to unregistered user

Explanation: Caused by a client trying to set ownership of the file to a user who is not registered in the proxy database.

User Action: None.

%UCX$-W-NFS_NOCMAP, Failed to find record in proxy database

Explanation: Client has no record in the proxy database.

User Action: Make entry in proxy database, if necessary.

%UCX$-F-NFS_NODLIM, Exceeded maximum host limit, not all clients can access server

Explanation: The amount of hosts in the proxy database exceeds the amount of hosts allowed by the parameter that can be set by the logical name UCX$NFS_HOST. Some of the hosts cannot access the server.

User Action: Change logical name value. Modify the logical UCX$NFS_HOST in the NFS startup file and restart the NFS server.

%UCX$-F-NFS_NOPROC, Unknown procedure in RPC message

Explanation: RPC request for the procedure is not supported by the NFS server.

User Action: Verify that the client supports the NFS software version.

%UCX$-F-NFS_NOPROG, Requested RPC program unavailable

Explanation: RPC request for the procedure is not supported by the NFS server.

User Action: Verify that the client supports the NFS software version.

%UCX$-W-NFS_NOSMAP, Failed to find Proxy record for default account to map supervisor

Explanation: Root client's UID and GID are changed by NFS process into values specified by UCX$00000000UID and UCX$00000000GID logical values. Proxy database must have a record to map the UID/GID into a VMS account. Record is not in proxy database, VMS account does not exist, or is disabled.

User Action: Check VMS account and add proxy record if necessary.

%UCX$-W-NFS_PORTNP, Remote service requested from nonprivileged port

Explanation: Security logical is set to disable nonprivileged ports from connecting to the NFS server and there is a message from the nonprivileged port.

User Action: Identify the client.

%UCX$-F-NFS_PROGVS, Invalid RPC program version number

Explanation: The RPC made a request for a program or program version which is not supported by the NFS server.

User Action: Determine if the client supports the correct version of the NFS software.

%UCX$-F-NFS_PRXEMP, PROXY database file is empty

Explanation: No information in the proxy database. No clients will be able to access the server.

User Action: Create a proxy database.

%UCX$-I-NFS_READDF, Failed to add account XXXX to volatile database

Explanation: The record was not added to the volatile database for one of the following reasons:

- The record already exists in the volatile database.
- The dynamic memory failed.
- The Connection failed to get the information for the account from the UAF file. Either the account was not in the UAF file or the account was disabled.

- The host names limit has been reached.
- Incorrect input data was used.

User Action: Depending on the reason why the add failed, do the following:

- If the record already exists in the volatile database, no action is necassary.
- If the dynamic memory allocation failed, provide enough memory for the process by either increasing the page file quotas in the NFS startup file or use SYSGEN to increase the virtual page count.
- If the account is not in the SYSUAF, you must add the account.
- If the account is disabled, you must enable the account.
- If the host names limit has been reached, you must increase the value for the host names parameter in the UCX$_NFSSTARTUP.COM and restart NFS.
- If incorrect input data was used, correct the information supplied by the parameters with the UCX command.

%UCX$-I-NFS_READDS, Added account XXXX to volatile database

Explanation: The record XXXX was added to the volatile database

User Action: None.

%UCX$-F-NFS_RPCPRS, Invalid RPC message format

Explanation: The client host is using a protocol different from the server.

User Action: Verify that the client supports the correct NFS protocol version.

%UCX$-I-NFS_SETGID, Setting default GID to XXX

Explanation: A SET NFS_SERVER/GID_DEFAULT=gid command was issued.

User Action: None.

%UCX$-I-NFS_SETITM, Setting inactivity timer to hh.mm

Explanation: A SET NFS_SERVER/INACTIVITY_TIMER=time command was issued.

User Action: None.

%UCX$-I-NFS_SETUID, Setting default UID to XXX

Explanation: A SET NFS_SERVER/UID_DEFAULT=uid command was issued.

User Action: None.

%UCX$-I-NFS_SHUTDW, NFS server is shutting down

Explanation: Server is shutting down.

User Action: None.

%UCX$-E-NFS_STALEF, Operation uses stale file handle

Explanation: The NFS client cached and used a file handle, but the file did not exist.

User Action: None.

%UCX$-E-STARTERROR, Error starting <service_name> service

Explanation: A START SERVICE command was executed, and the specified service is already active.

User Action: None.

%UCX$-E-STOPERROR, Error stopping <service_name> service

Explanation: A STOP SERVICE command was executed, and the specified service was not active.

User Action: None.

%UCX$-E-SUBNAM, Substitute hostname XXXX for hostname

Explanation: An NFS client used an alias host name that was not in the volatile proxy database. NFS updates the volatile database.

User Action: None.

%UCX$-F-NFS_UIDMAX, Invalid UID range

Explanation: Maximum UID value of 65535 is exceeded by the client.

User Action: Enter an acceptable UID value.

%UCX$-I-NFS_UMASUC, All file systems are unmounted

Explanation: Infomational.

User Action: None.

%UCX$-I-NFS_UMNSUC, Unmounted file system XXXXXX

Explanation: Informational.

User Action: None.

%UCX$-W-NFS_USRMNT, User-level mount request rejected

Explanation: Security logical is set to disable user level mounts on the server, and an attempt was made to do a mount.

User Action: Change security logical value if necessary.

%UCX$-W-NFS_VDBREC, NFS proxy account XXXXXX disabled, client record ignored

Explanation: A user's account on the VMS is disabled. This user will not be able to access NFS server.

User Action: Enable VMS account if necessary.

%UCX$-I-UCP_ACPQIO, Failure on Internet ACP QIO

Explanation: An interface operation requiring the INET_ACP has failed.

User Action: Review the interface command and reenter with corrected information.

%UCX$-I-UCP_ADR_HOST, Internet address specifies a host

Explanation: The Internet address specifies a host address when a network address is required.

User Action: Reissue the command with the network address.

%UCX$-I-UCP_ADR_NETW, Internet address specifies a network

Explanation: The Internet address specifies a network address when a host address is required.

User Action: Reissue the command with the host address.

%UCX$-I-UCP_ALIAS, Record is an alias

Explanation: The host or network information returned from the database came from an alias of a host or network entry.

User Action: None.

%UCX$-E-UCP_ARPERROR, Error processing ARP request

Explanation: An error was detected while processing an ARP command.

User Action: Review the documentation for the ARP command and reenter the command.

%UCX$-E-UCP_BCPMBX, Cannot open Connection data mailbox

Explanation: Connection could not open a temporary mailbox that it uses to receive data from NFS and the Connection file system.

User Action: Check for the required process privileges and process quotas.

%UCX$-E-UCP_BINDERROR, Error processing BIND or UNBIND request

Explanation: An error was detected while processing a BIND or UNBIND command.

User Action: If the error is a duplicate name, then perform a SHOW BIND command for a list of bound file systems. Review the documentation for the BIND or UNBIND command, and reenter the command.

%UCX$-E-UCP_BINDNOCFS, Container directory is not bound

Explanation: An operation involving a container directory is being attempted, and the container directory has not been bound.

User Action: Execute a BIND command for the container directory.

%UCX$-E-UCP_BINDNOSYS, File system is not bound

Explanation: An operation involving a file system, which is in a container directory or on a VMS disk, is being attempted, and the file system has not been bound.

User Action: Execute a BIND command for the file system.

%UCX$-E-UCP_BINDNOVMS, Target VMS disk volume is not bound

Explanation: An operation involving a container file is being attempted, and the VMS disk where the container directory resides has not been bound.

User Action: Execute a BIND command for the VMS disk on which the container directory is to reside.

%UCX$-F-UCP_BUGCHK, Connection internal error

Explanation: Internal error in the Connection code.

User Action: Submit an SPR providing the following information:

- Describe the command that was entered.
- Save the database files, if the command involved one or more databases.

%UCX$-E-UCP_CFSERROR, Error processing Connection file system request

Explanation: An error was detected in an operation requiring a call to the Connection file system.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_CMDERR, NFS command error

Explanation: NFS does not have the functionality to process a command sent by Connection.

User Action: Check to see that the Connection and NFS processes were derived from the same Connection release.

%UCX$-E-UCP_COMERR, NFS server has not created a command mailbox

Explanation: NFS is still in an initialization state. It has not created a mailbox to which Connection can issue commands.

User Action: Wait until NFS has completed initialization and reissue the command.

%UCX$-E-UCP_DATAPROB, Error accessing the XXXX database

Explanation: There was an error in accessing the XXXX database.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_DELINTER, Error deleting interface: XXXX

Explanation: The specified interface could not be deleted.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-I-UCP_DIRECTORY, Pathname resolves to a directory

Explanation: A pathname that is a directory was specified for a CFS operation that requires a data file as an object.

User Action: Perform a DIRECTORY on the file system to locate the target file.

%UCX$-I-UCP_DRIVERQIO, Failure on Internet driver QIO

Explanation: An interface operation that is handled by the Internet driver has failed.

User Action: Review the interface command and reenter with corrected information.

%UCX$-W-UCP_DUPALLHOST, File system already exported to all hosts

Explanation: A new record was added to the export database; however, the file system was already exported to all hosts.

User Action: None required, but you may want to review the export database entries for this file system to eliminate duplicate entries.

%UCX$-W-UCP_DUPHIEXPO, File system already exported at higher level

Explanation: A new record was added to the export database; however, the file system was already exported to the host at a higher level. For example the new record specified /a/b, when /a was present in the database.

User Action: None required, however, you may want to review the export database entries for this file system to eliminate duplicate entries.

%UCX$-E-UCP_DUPHOST, Duplicate UCX$HOST entry for XXXX

Explanation: Information specified in a SET HOST command is already present in the host database.

User Action: None.

%UCX$-W-UCP_DUPHOSTEXPO, File system already exported at specific host

Explanation: A new wildcarded host record was added to the export database; however, the file system was already exported to a specific host.

User Action: None required, but you may want to review the export database entries for this file system to eliminate duplicate entries.

%UCX$-W-UCP_DUPLOEXPO, File system(s) already exported at lower level

Explanation: A new record was added to the export database; however, a file system was already exported to the host at a lower level. For example, the new record specified /a, when /a/b was present in the database.

User Action: None required, however, you may want to review the export database entries for this file system to eliminate duplicate entries.

%UCX$-E-UCP_DUPNETWORK, Duplicate UCX$NETWORK entry for XXXX

Explanation: Information specified in a SET NETWORK command is already present in the network database.

User Action: None.

%UCX$-E-UCP_DUPRECORD, Duplicate XXXXX information

Explanation: You attempted to enter a record that has duplicate key information.

User Action: Issue a SHOW command for the particular database to obtain a list of entries.

%UCX$-E-UCP_ETCHOST, Error processing /etc/hosts file

Explanation: An ULTRIX-formatted */etc/hosts* file could not be accessed while processing a CONVERT/xxxx HOST command.

User Action: Review the documentation for the CONVERT command.

%UCX$-E-UCP_ETCNETWORK, Error processing /etc/networks file

Explanation: An ULTRIX-formatted */etc/networks* file could not be accessed while processing a CONVERT/xxxx NETWORK command.

User Action: Review the documentation for the CONVERT command.

%UCX$-E-UCP_ETCPASSWD, Error processing /etc/passwd file

Explanation: An ULTRIX-formatted */etc/passwd* file could not be accessed while processing a CONVERT proxy command.

User Action: Review the documentation for the CONVERT command.

%UCX$-I-UCP_EXPORTADD, Export information added to database

Explanation: Information was added to the export database although there were some warning messages.

User Action: None.

%UCX$-E-UCP_EXPORTERROR, Error processing EXPORT request

Explanation: An error was detected in an operation requiring an access to the export database.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-I-UCP_FILE, Pathname resolves to a file

Explanation: A pathname that is a data file was specified for a CFS operation that requires a directory as an object.

User Action: Perform a DIRECTORY on the file system to locate the target directory.

%UCX$-W-UCP_FLDTRN, Field value has been truncated : XXXXX

Explanation: The specified string or integer value is greater than the maximum field size.

User Action: Make the specified string or integer value smaller than the maximum field size.

%UCX$-E-UCP_HOSTERROR, Error processing HOST request

Explanation: An error was detected in an operation requiring an access to the host database.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_IMPLEMENT, This service has not been implemented

Explanation: You issued a command that is not supported.

User Action: Review the Connection commands to determine the set of services for the release.

%UCX$-E-UCP_INETERROR, Error accessing Internet

Explanation: An error was detected in an operation requiring access to Connection Internet software.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_INSARP, Insufficient ARP information

Explanation: You attempted to add an ARP table entry, but not enough information was provided.

User Action: Review the documentation for SET ARP.

%UCX$-E-UCP_INSKE, Insufficient information for XXXXX record

Explanation: You attempted to enter a record into a database, but did not specify enough information.

User Action: Reissue the command with the necessary qualifiers.

%UCX$-E-UCP_INTEERROR, Error defining or modifying interface

Explanation: An error was detected in specifying an Internet interface.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_INVADR, Invalid Internet address: XXXXX

Explanation: The Internet address specified has an invalid format.

User Action: Refer to Chapter 3 for an explanation of valid Internet addresses.

%UCX$-E-UCP_INVBRDMASK, Invalid broadcast mask

Explanation: There is a syntax error in the specification of an Internet broadcast mask.

User Action: Refer to Chapter 2 for a description of an Internet broadcast mask.

%UCX$-E-UCP_INVCLUST, Invalid cluster definition for interface: XXXX

Explanation: The cluster host for the interface could not be defined.

User Action: Review the documentation for the SET INTERFACE command.

%UCX$-E-UCP_INVDEVNAME, Invalid Internet device name

Explanation: The device name specified is not valid.

User Action: Refer to Chapter 3 for a list of valid Internet device names.

%UCX$-E-UCP_INVDIR, Error referencing VMS directory

Explanation: There is a problem in translating a VMS directory specification. Typically, this error is caused from a logical name being the same as the directory name.

User Action: Examine the directory specification for errors. If this error occurred when a pathname was specified, issue a SHOW BIND command to examine the VMS directory specification.

%UCX$-E-UCP_INVDISK, Invalid VMS disk specification

Explanation: An incorrect syntax has been specified for a VMS disk device.

User Action: Perform a VMS SHOW DEVICE command for a list of valid VMS disk names.

%UCX$-E-UCP_INVETHER, Invalid Ethernet address: XXXXX

Explanation: The Ethernet address specified has an invalid format.

User Action: Reissue the command with an Ethernet address of the format xx-xx-xx-xx-xx-xx, where xx is a 2-character hexadecimal value.

%UCX$-E-UCP_INVETHERNAM, Invalid Ethernet name

Explanation: This is a UCX$UCP internal error.

User Action: Submit an SPR to Digital with a copy of the command that was used when the error occurred. Also include a listing from a SHOW INTERFACE command.

%UCX$-E-UCP_INVGATEWAY, Gateway specifies a network

Explanation: The gateway for a route specified via a SET ROUTE /GATEWAY or a SET ROUTE /G_ADDRESS specifies a network address.

User Action: Reissue the SET ROUTE command with a gateway that specifies a host address.

%UCX$-E-UCP_INVINTER, Error defining interface: XXXX

Explanation: A problem was encountered while trying to process an interface request.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_INVINTERNAM, Invalid interface name

Explanation: The specified interface name is not valid.

User Action: Review the list of supported Connection interface device names.

%UCX$-E-UCP_INVKEY, Invalid information for XXXXX file

Explanation: A value for a key was not in an acceptable range of values.

User Action: Refer to the Connection database documentation for a description of key fields. Refer to the documentation of the Connection command.

%UCX$-E-UCP_INVNET, Invalid network number: XXXX

Explanation: There is a syntax error in the specification of an Internet network number, probably because the Internet address specifies a host.

User Action: Refer to Chapter 2 for a description of network numbers.

%UCX$-E-UCP_INVNETADDR, Invalid Internet address

Explanation: There is a syntax error in the specification of the Internet address.

User Action: Refer to Chapter 2 for a description of Internet addresses.

%UCX$-E-UCP_INVNETFLAG, Invalid interface flags

Explanation: An illegal combination of flags was entered with a SET INTERFACE command.

User Action: Review the documentation for the SET INTERFACE command.

%UCX$-E-UCP_INVNETMASK, Invalid network mask

Explanation: There is a syntax error in the specification of an Internet network mask.

User Action: Refer to Chapter 2 for a description of an Internet network mask.

%UCX$-E-UCP_INVPARM, Invalid parameter value for: XXXXX

Explanation: A parameter value from a command line could not be processed.

User Action: Refer to Chapter 5 for the description of the specific Connection command and reissue the command with the correct parameter value.

%UCX$-E-UCP_INVPATH, Invalid pathname

Explanation: There is a syntax error in the specified pathname.

User Action: Correct the syntax of the pathname and resubmit the command.

%UCX$-E-UCP_INVQUAL, Invalid qualifier value for /XXXXX

Explanation: A qualifier value from a command line could not be processed.

User Action: Refer to Chapter 4 for the description of the specific Connection command and issue the command with the correct qualifier.

%UCX$-E-UCP_INVRECORD, Invalid record for XXXXX file

Explanation: An RMS error has occurred while trying to access a database file.

User Action: The error message contains the RMS status. Refer to the *VAX Record Management Services Reference Manual* for the corrective procedure.

%UCX$-E-UCP_INVSOCKNAM, Invalid socket name

Explanation: There is a syntax error in the socket specification.

User Action: Correct the syntax of the device_socket name and reissue the command. You may want to execute a SHOW DEVICE command to obtain a list of device_sockets on your system.

%UCX$-I-UCP_LOOPACT, XXXX is alive

Explanation: The LOOP operation to the specified host is successful and the host has responded.

User Action: None.

%UCX$-E-UCP_LOOPERROR, Error processing LOOP request

Explanation: A problem was encountered while trying to process a LOOP request.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-I-UCP_LOOPINACT, XXXX does not respond

Explanation: The LOOP operation to the specified host is successful, but the remote host has not responded.

User Action: None.

%UCX$-E-UCP_MISROUTE, ROUTE database mismatch

Explanation: An inconsistency exists in the route database, normally due to an entry in the Route database which specifies a host or network that is no longer present in the Internet database.

User Action: Perform a SHOW ROUTE command to determine the problem.

%UCX$-E-UCP_NETWORKERROR, Error processing NETWORK request

Explanation: An error was detected in an operation requiring an access to the network database.

User Action: Other messages will be printed with this error. Review these messages to determine the problem.

%UCX$-E-UCP_NFSERROR, Error communicating with NFS

Explanation: A problem was encountered while trying to communicate with the NFS process.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_NFSMBX, Cannot open NFS command mailbox

Explanation: Connection has insufficient privilege to find information about the NFS server.

User Action: Check the process privileges and the process quotas.

%UCX$-W-UCP_NOARP, No ARP entries found

Explanation: There is no ARP entry for the specified host. If the specified host is not currently accessing the local host, no ARP entry will be present.

User Action: None.

%UCX$-W-UCP_NOAUTOROUT, Automatic route not found

Explanation: An interface's Internet address is being modified and there is no automatic route record present in the route database.

User Action: No action is necessary; however, this condition signals the possibility that the route database is corrupted.

%UCX$-E-UCP_NOCONTAINER, Container file not found

Explanation: The specified container file does not reside in the named directory.

User Action: Issue a SHOW BIND command to find the container file's VMS directory.

%UCX$-E-UCP_NOCONVERT, Specify /ULTRIX or /VMS

Explanation: A CONVERT command was issued without specifying /ULTRIX or /VMS.

User Action: Review the documentation for the CONVERT command.

%UCX$-E-UCP_NOCREATE, Could not create database

Explanation: A problem was encountered while trying to create a Connection database file.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_NODESTNAM, Destination name is not defined

Explanation: The host or network name that was specified with a SET ROUTE command has not been defined in the host or network database.

User Action: Perform the following:

1 Define the host name with a SET HOST command.

2 Define the network name with a SET NETWORK command.

3 Issue a SET ROUTE /ADDRESS command.

%UCX$-W-UCP_NODEVLOCL, Local device not found

Explanation: A LOOP command is being requested and the local interface is not defined.

User Action: Execute the command procedure SYS$MANAGER:UCX$INET_SET_INTERFACES.COM in order to define the local host.

%UCX$-E-UCP_NOFILE, Cannot find XXXXX file

Explanation: The logical name that points to a Connection database has not been defined, or a logical name points to a database file specification that has no corresponding file.

User Action: Check all database logical name definitions. Refer to the *VMS / ULTRIX Connection Installation Guide* for a list of logical database file names.

%UCX$-E-UCP_NOFUNCTION, Unsupported function

Explanation: CFS does not have the functionality to process a command sent by the Connection.

User Action: Check to see that the Connection and NFS processes were derived from the same Connection release.

%UCX$-E-UCP_NOGATENAM, Gateway name is not defined

Explanation: The host name specified with a SET ROUTE /GATEWAY command but has not been defined in the host database.

User Action: Define the host name with a SET HOST command, or issue a SET ROUTE /G_ADDRESS command.

%UCX$-E-UCP_NOINET, The Internet driver is not loaded

Explanation: You attempted to perform an Internet command, and the Internet software has not been installed or activated.

User Action: Activate or install the Internet software. Refer to the *VMS/ULTRIX Connection Installation Guide* for the Internet installation procedure.

%UCX$-E-UCP_NONFS, NFS server is not active

Explanation: The NFS server has not been started.

User Action: Start the NFS server.

%UCX$-E-UCP_NOPATH, No files found

Explanation: No files were present for the specified DIRECTORY command.

User Action: Perform a wildcarded DIRECTORY command to obtain a list of files present in the file system.

%UCX$-E-UCP_NOPRIV, Insufficient privilege for xxxxx command

Explanation: Insufficient privilege for xxxxx command.

User Action: Check the command documentation for a list of required privileges.

%UCX$-E-UCP_NORESPOND, NFS server is not responding

Explanation: One of the following has happened:

- The activity on the system is such that the NFS server has not responded to the Connection command before Connection's communication timer has expired. This is expected when the server is busy processing NFS requests.
- The NFS process was stopped during the execution of a Connection command.

User Action: Perform one of the following actions:

- If the NFS server is still active, reissue the command.
- If the NFS server has been stopped, restart the NFS server using the NFS installation procedure.

%UCX$-W-UCP_NORECORD, XXXXX record not found

Explanation: The information requested does not exist in a database where XXXXX is the logical name for the database file.

User Action: Reenter the command.

%UCX$-E-UCP_NOSERVICE, No such service

Explanation: The service specified to an NFS command is not known to the NFS process.

User Action: Issue a SHOW NFS /SERVICE command for a list of supported NFS services.

%UCX$-E-UCP_NOTSTARTED, The Internet software has not been started

Explanation: A command that requires active Connection Internet software was attempted.

User Action: Execute the command procedure SYS$MANAGER:UCX$INET_STARTUP.COM to start the Connection Internet software.

%UCX$-E-UCP_NOUSER, No such user

Explanation: A SHOW NFS /USERNAME command was issued with a specified user who is not known to the NFS process.

User Action: Reenter with the proper name.

%UCX$-I-UCP_NOVMSUSER, VMS user: XXXXX does not exist

Explanation: The VMS username for the proxy record was added with the ADD PROXY command but has not been defined in the VMS UAF file.

User Action: Define the VMS username with the VMS authorize utility.

%UCX$-I-UCP_PARENT, Record is a parent

Explanation: The host or network information returned from the database came from an original host or network entry (not an alias).

User Action: None.

%UCX$-E-UCP_PROTERR, Unsupported protocol

Explanation: There is a communication protocol mismatch between Connection and NFS processes. This may occur when the Connection and NFS processes are derived from dissimilar releases.

User Action: Check to make sure that the Connection and NFS processes come from similar Connection releases.

%UCX$-I-UCP_PROXYADD, Proxy information added to database

Explanation: Information was added to the proxy database although there were some warning message(s).

User Action: None.

%UCX$-E-UCP_PROXYERROR, Error accessing proxy database

Explanation: An error was detected in an operation requiring an access to the proxy database.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_ROUTEERROR, Error accessing ROUTE database

Explanation: An error was detected in an operation requiring an access to the route database.

User Action: Other messages will be printed with this error. Review these messages to correct the problem.

%UCX$-E-UCP_SEQERR, NFS transaction sequence error

Explanation: There is an internal communication message mismatch between Connection and NFS processes.

User Action: Reissue the last command and if the problem persists, rerun the Connection.

%UCX$-E-UCP_STARTED, The Internet software has been started

Explanation: A command that requires that the Connection Internet software to be inactive was attempted.

User Action: Execute the command procedure SYS$MANAGER:UCX$INET_SHUTDOWN.COM to stop the Connection Internet software.

A.2 File System Messages Directed to the Operator's Console

This section contains file system error messages that are directed to the operator's console by default. If the messages cannot be written to the operator's console, they are directed to the device defined by the logical name UCX$CFS_FATAL_MESSAGES. This logical name must point to a device, because, the messages are written using $QIO system service calls.

%UCX$-E-CFS_ATTRFAIL, Failed to [read/write] file attributes

Explanation: The file system attempted to read or write a file's attributes using the appropriate Files–11 QIO and it was returned an error which it did not expect.

%UCX$-E-CFS_DATAFAIL, Failed to [read/write] file data

Explanation: The file system attempted to read or write virtual blocks of a file and it was returned an error which it did not expect.

%UCX$-E-CFS_DEACCESS, Files to deaccess file

Explanation: The file system was returned an error when it attempted to deaccess a file using the appropriate Files–11 QIO.

%UCX$-E-CFS_DASSGN, Failed to deassign channel

Explanation: The file system was returned an error when it attempted to deassign a channel as part of deaccessing a file.

%UCX$-E-CFS_DELETEINO, Failed to delete inode

Explanation: The ULTRIX file system attempted to delete an inode within the file system's container file, and the underlying file system routines returned an unexpected error.

%UCX$-E-CFS_ERROR, Error: <error text>

Explanation: This message is used to return error information that does not indicate a problem with the file system. Rather, this information is of general use to the system manager in determining the cause of odd behavior.

%UCX$-E-CFS_INVFSDEF, Incompatible file system definition

Explanation: A file system definition in the UCX$CFS_* logical name tables is either from a previous release, or was placed into the table without using UCX.

%UCX$-E-CFS_IVCHAN, Invalid CFS channel number

Explanation: A problem was detected when processing an internal ULTRIX file system channel.

%UCX$-E-CFS_LCKDEQ, Attempt to $DEQ lock failed, lock ID <hex-lock-ID>

Explanation: An unexpected error was returned when the file system attempted to $DEQ a resource lock.

%UCX$-E-CFS_LCKENQ, Attempt to $ENQ lock failed

Explanation: An unexpected error was returned when the file system attempted to $ENQ a resource lock. Often this message indicates that a process quota is too low.

%UCX$-E-CFS_LCKLCKD, Attempt to lock resource previously locked

Explanation: An attempt to lock a resource on which a lock is already held by the current file system operation was detected.

%UCX$-E-CFS_LCKNCVT, Attempt to $ENQ lock from <lock-mode> to <lock-mode> failed

Explanation: An attempt to change the lock mode of a particular resource has failed. The starting and requested modes are shown.

%UCX$-E-CFS_ULCKNLCKD, Attempt to release resource not previously locked

Explanation: An attempt to unlock a resource on which no lock is currently held for the current file system operation was detected.

%UCX$-E-CFS_UNXBLKAST, Unexpected blocking AST received for type <0-4>

Explanation: A blocking AST was received indicating that the file system's notion of a resource lock's state is not consistent with the system's view.

%UCX$-E-CFS_UNXQIOSTS, Unexpected $QIO status returned

Explanation: An unexpected return status was detected for a $QIO. This message is generated purely for diagnostic purposes and does not indicate a problem in the file system.

%UCX$-E-CFS_VMSDIRFMT, Inconsistent format within VMS directory file

Explanation: The contents of a VAX/VMS ODS2 directory does not appear to be valid. This message is generated to warn the system manager that the file system attempted to read a directory (NFS READDIR) and the directory did not contain valid contents.

%UCX$-E-CFS_VMSDIRINI, Failed to initialize VMS directory contents

Explanation: Part of the VAX/VMS directory creation code failed. In particular, the directory entry has been created, the file header initialized, but the initial data blocks of the directory were not written properly.

%UCX$-E-CFS_PRCEXIT, Process exit forced due to unrecoverable condition

Explanation: The file system code executing in EXECUTIVE mode detected a condition from which it could not recover. It deemed the situation severe enough to terminate the process. It generates this message as a courtesy to the system manager, to explain the reason for the disappearance of a process.

The following messages may appear in conjunction with the above messages to provide additional information about the particular problem which is being reported.

%UCX$-I-CFS_BIND, File system <file-spec> bound as <path-spec>

%UCX$-I-CFS_CHANNEL, Channel number <hex-number>

%UCX$-I-CFS_CLUSTER, Virtual blocks <start> through <end>

%UCX$-I-CFS_CONTAINERID, Container <device-file-ID-spec>

%UCX$-I-CFS_FHANDLE, File handle <hex-file-handle>

%UCX$-I-CFS_FILENAME, File name <file-name-string>

%UCX$-I-CFS_FILESPEC, File specification <file-spec-string>

%UCX$-I-CFS_FILESYSTEM, File system <file-system-string>

%UCX$-I-CFS_FILEID, File <device-file-ID-spec>

%UCX$-I-CFS_INODE, Inode number <hex-number>

%UCX$-I-CFS_LOCKSTS, Lock mode <lock-mode>, <number>/<number>
granted status <hex-number>

%UCX$-I-CFS_OFFSET, File offset <hex-number>, length <hex-number>

%UCX$-I-CFS_PATHNAME, Pathname <path-name-string>

%UCX$-I-CFS_THREAD, Service <service-name-string>, sequence <number>

%UCX$-I-CFS_RDCB, Resource descriptor address <hex-number>, type <0-4>

%UCX$-I-CFS_RESOURCE, Resource is <resource-type-string>

%UCX$-I-CFS_UNBIND, File system <path-name-string> is unbound

%UCX$-I-CFS_IVRESSTS, Invalid resource state

%UCX$-E-UFS_BAD_ID, CF packid mismatch

Explanation: The ULTRIX file system pack identification does not match the VMS disk volume

User Action: Run ANALYZE CONTAINER to correct the identification.

%UCX$-E-UFS_BHDRETCNT, Failed to perform CAHE while processing container file. Container FID=(XX,XX,XX); return count was incorrect (should be 1).

Explanation: An internal CFS error in performing the CACH operation occurred while processing the container file (FID=(XX,XX,XX).

User Action: Run ANALYZE/CONTAINER on the specified container file.

%UCX$-E-UFS_BITCLEAR, Bitmap bit to be cleared already cleared: base BBB, position PPP, bitno NNN

Explanation: An internal CFS error has occurred in performing bitmap clear bit operation: inconsistent bitmap information encountered.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_BITOPFAIL, To perform BITOP; operation: XXX, status STATUS

Explanation: An internal CFS error in performing bitmap operation XXX has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_BITSET, Bitmap bit to be set already set: base BBB, position PPP, bitno NNN

Explanation: An internal CFS error has occurred in performing bitmap set bit operation: inconsistent bitmap information encountered.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_BMEXTFAIL, Failed to extend container file, status STATUS

Explanation: The ULTRIX file system is full and its size has reached the maximum size limit, or internal CFS error that prevents the ULTRIX file system to be extended.

User Action: Examine the error message and take appropriate action. If this messages is preceded by a %UCX$-E-UFS_NOROOM message, see user action for that message. Otherwise, submit an SPR with all the information describing the condition under which this message was generated.

%UCX$-E-UFS_BMOUTCF, Bitmap operation extends beyond existing cells (XXX > YYY)

Explanation: An internal CFS error in performing bitmap operation. Bytes range requested pass file system boundary.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_CACHEFAIL, Failed to perform CACHE, operation XXX, status STATUS

Explanation: An internal CFS error in performing disk I/O.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_CGIOERROR, Failed to complete CG_IO: XXX

Explanation: An internal CFS error in performing cell group I/O to disk.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_CGTOOBIG, Failed to create cell group; required length (LLL) exceeds maximum (MMM)

Explanation: Unable to create a cell group; the required cell group length cannot be satisfied.

User Action: Examine the error message and take appropriate action. If this message is not caused by a ULTRIX file system being full, submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_CGTYPERR, Cell NNN type is not TYPE1, but TYPE2

Explanation: An internal CFS error. Cell number NNN is type TYPE2, not the expected type TYPE1.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_DEALLCFAIL, Failed to deallocate XXX cells; status STATUS

Explanation: An internal CFS error in deallocating XXX number of cells has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_DIREXTFAIL, Failed to extend directory cell group

Explanation: Unable to extend a directory cell group.

User Action: Examine the error message and take appropriate action. If this message is not caused by a ULTRIX file system being full, submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_DOBITFAIL, Failed to set/clear bit in bitmap; status STATUS

Explanation: An internal CFS error in performing bit set/clear operation has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_ENTRYOPERR, Failed to complete ENTRYOP

Explanation: An internal CFS error in performing directory entry operations.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_EXTBMPFAIL, Failed to create bitmap cell group

Explanation: An internal CFS error in allocating bitmap cell groups has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_FRDCBFAIL, Failed to create RDCB; object XXX, status STATUS

Explanation: An internal CFS error in creating resource descriptor control block for the object XXX has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_INVOPER, Invalid operation requested: XXX

Explanation: An internal CFS error. Invalid directory entry operation was requested to module XXX.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_INVRDCB, Invalid RDCB: XXX

Explanation: An internal CFS error. The resource descriptor control block indicates the object has changed is invalid.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_INVSIZE, Invalid size specified: XXX

Explanation: An internal CFS error. Invalid size has been specified for object XXX.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_INVTYPE, Invalid type specified: XXX

Explanation: An internal CFS error. Invalid type has been specified for object XXX.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_LOCKFAIL, Failed to set lock on XXX RDCB

Explanation: An internal CFS error in creating a lock on the resource descriptor control block for object XXX has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-W-UFS_NOROOM, ULTRIX file system is full

Explanation: We need to provide an explanatin here.

User Action: Remove unused files from the ULTRIX file system to free up space, or start using a new file system.

%UCX$-E-UFS_NOTFOUND, Failed to find XXX

Explanation: An internal CFS error occurred. The object XXX was not found.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_RELDRDCBF, Failed to release XXX RDCB, status STATUS

Explanation: An internal CFS error in deallocating a resource descriptor control block for object XXX has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

%UCX$-E-UFS_RELOCKFAIL, Failed to release lock on XXX RDCB, status STATUS

Explanation: An internal CFS error in releasing a lock on the resource descriptor control block for object XXX has occurred.

User Action: Submit an SPR with all the information describing the conditions under which this message was generated.

A.3 Internet Messages Directed to the Operator's Console

The following Internet error messages are directed to the operator's console by default.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet ACP Tracing RIP Packets,
REQUEST from <host_address>: <routing_information,

Explanation: A request for routing information was received from the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user INTERnet on <host_name>,
INTERnet ACP Tracing RIP Packets,
RESPONSE from <host_address> <routing_information>,

Explanation: A response for routing information was received from the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Start Dynamic Routing,

Explanation: The dynamic routing server was started on the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Stop Dynamic Routing,

Explanation: The dynamic routing server was stopped on the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet Loaded,

Explanation: The Internet driver and ACP were loaded.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet Started,

Explanation: The Internet driver was started by some one issuing the UCX command START COMMUNICATION.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Recover Abnormal Shutdown,

Explanation: The Internet ACP detected a previous abnormal Internet shutdown. The Internet ACP attempts to recover.

User Action: Use SYS$MANAGER:UCX$SHUTDOWN or SYS$MANAGER:UCX$INET_SHUTDOWN to shutdown the Internet.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Abnormal Exit,

Explanation: The Internet ACP detected an abnormal condition and it exits.

User Action: Submit an SPR with a crash dump of your system.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP enabled VMS Internet cluster on: <interface_name>,

Explanation: The Internet ACP was enabled.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP failed to set the Cluster Timer Status: <error_code>,

Explanation: The Internet ACP failed to set the cluster time.

User Action:

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP failed to start the Internet Cluster Status: <error_code>,

Explanation: The Internet failed to start the Internet cluster status.

User Action:

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Lost Sync on <ethernet_device_name> Ethernet Channels,
Status = <error_code>,

Explanation: The Ethernet controller detected an error condition. The Internet will attempt to restart the Ethernet channels. Failure or success to restart the Ethernet channels is logged into the operator log file.

User Action:

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Restarted <ethernet_device_name> Ethernet Channels,

Explanation: The Ethernet channels assigned to Internet were successfully restarted.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Restart once more <ethernet_device_name> Ethernet Channels,

Explanation: An attempt to restart the Ethernet channels assigned to Internet failed. Another attempt is being made.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Restart <ethernet_device_name> Ethernet Channels Failed,

Explanation: All attempts to restart the Ethernet channels assigned to Internet failed. No other attempt is being made. The Internet interface has been deleted.

User Action: You can attempt to recreate the interface at a later time.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Remote Terminal Services START - RLOGIN,

Explanation: rlogin has been started on the specified host.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Remote Terminal Services STOP - RLOGIN,

Explanation: rlogin has been stopped on the specified host.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
RLOGIN Login Request from Remote Host: <host_name> Port: <port_number>,

Explanation: A user has logged in to the specified host using rlogin.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
RLOGIN Logout Request from Remote Host: <host_name>, Port: <port_number>

Explanation: A user has logged out of an rlogin session on the specified host.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Created Internet interface: <interface_name>,

Explanation: This is an informational message. The specified Internet interface has been created on the specified host.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Deleted Internet interface: <interface_name>,

Explanation: This is an informational message. The specified Internet interface has been deleted from the specified host.

User Action: None.

%%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%%,
Message from user Internet on <host_name>,
Internet ACP Remote Terminal Services START — TELNET,

Explanation: TELNET has been started on the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host name>,
Internet ACP Remote Terminal Services STOP — TELNET,

Explanation: TELNET has been stopped on the specified host.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host name>,
TELNET Login request from Remote Host: <host_address> Port: <port_num>,

Explanation: A user has logged in to the specified host using TELNET.

User Action: None.

%%%%%%%%%% OPCOM <date> <time> %%%%%%%%%%,
Message from user Internet on <host name>,
TELNET Logout request from Remote Host: <host_address> Port: <port_num>,

Explanation: A user has logged out of a TELNET session on the specified host.

User Action: None.

B

ANALYZE CONTAINER Validation Messages

ANALYZE CONTAINER produces two types of messages: informational and error messages. Informational messages describe the current processing of the analysis. Error messages describe problems found in the container file and the Connection ULTRIX file system.

Usually error messages appear in two parts. The first part describes the problem, and the second part describes the solution ANALYZE CONTAINER will use to correct the problem. For some problems, if you have specified both the REPAIR and CONFIRM options, ANALYZE asks you to confirm whether it should implement the correction or not. Most of the problems do not require confirmation, even when the CONFIRM option is set, because they are related to errors found in the internal structuring of the container file. Usually these problems are caused by a hardware or software problem, or the system has been brought down while the file system was active. If you suspect any malfunction in the software and have SPR service, please submit an SPR with the container file plus the related VMS directories files associated with the Connection ULTRIX file system described by the following command:

```
$ BACKUP disk:[container_dir...]*.*;* SPR/SAVED
```

Container_dir is the directory specification of the directory where the container file is located.

In the message text, items surrounded by angle brackets (< >) denote values to be calculated at run time.

B.1 ANALYZE CONTAINER Summary Information

The following messages are printed at the end of the ANALYZE CONTAINER analysis. They provide information on the use of the Connection ULTRIX file system.

```
Analysis Summary Report
=======================

Cell size: n bytes
Container size: n cells (n bytes)
Container usage: n cells used (n%)
Number of errors detected:
Number of corrections made:
Lost+found changed/created:
Number of directories:
Number of regular files:
Number of symbolic files:
Number of imported files:
```

B.2 General Information Messages

The following messages are informational and do not require any user action. They indicate the progress of the analysis process. These messages always appear in the order they are listed here.

%UCX$-I-ANA_STARTSUPER, Starting super block validation

Explanation: ANALYZE/CONTAINER is starting to check the super block.

User Action: None.

%UCX$-I-ANA_ENDSUPER, End of super block validation

Explanation: ANALYZE/CONTAINER has successfully finished checking the super block.

User Action: None.

%UCX$-I-ANA_STARTINODE, Starting inode validation

Explanation: ANALYZE/CONTAINER is starting to check the inodes.

User Action: None.

%UCX$-I-ANA_ENDINODE, End of inode validation

Explanation: ANALYZE/CONTAINER has successfully finished checking the inodes.

User Action: None.

%UCX$-I-ANA_STARTDIR, Starting directory validation

Explanation: ANALYZE/CONTAINER is starting to check the directories.

User Action: None.

%UCX$-I-ANA_ENDDIR, End of directory validation

Explanation: ANALYZE/CONTAINER has successfully finished checking the directories.

User Action: None.

%UCX$-I-ANA_STARTBITMAP, Starting bitmap validation

Explanation: ANALYZE/CONTAINER is starting to check the Connection file system space allocation.

User Action: None.

%UCX$-I-ANA_ENDBITMAP, End of bitmap validation

Explanation: ANALYZE/CONTAINER has successfully finished checking the Connection file system space allocation.

User Action: None.

%UCX$-I-ANA_STARTFIX, Validating the directory hierarchy

Explanation: ANALYZE/CONTAINER is starting to correct any directory hierarchy problem and to place disconnected files and directories in the lost and found directory.

User Action: None.

%UCX$-I-ANA_ENDFIX, End validating the directory hierarchy

Explanation: ANALYZE/CONTAINER has successfully finished correcting any directory hierarchy problem.

User Action: None.

%UCX$-I-ANA_QUIT, Unable to continue analysis

Explanation: ANALYZE/CONTAINER is unable to complete the analysis due to fatal errors found in the container file.

User Action: See other error messages generated by ANALYZE/CONTAINER to determine the appropriate action.

%UCX$-I-ANA_ENDSUCCESS, End of analysis

Explanation: ANALYZE/CONTAINER has successfully completed the analysis.

User Action: None.

B.3 Superblock Validation Messages

The following error messages are related to errors detected in the superblock. Generally, these problems are corrected automatically (if the REPAIR option is set) without user confirmation or action. All other errors indicate data corruption caused by a hardware or software problem, or the system was shut down abruptly while the Connection file system was active. If you believe the problem was due to software problems, send an SPR.

%UCX$-E-ANA_SUP_BADBCGSIZE, Problem: Bad bitmap cell group size: <bad value>

Solution: Will be replaced by default value (<good value>)

Explanation: Super block contains invalid value for the initial bitmap cell group allocation size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADBITMAP, Problem: Invalid Bitmap inode number: <bad inode number>

Solution: Will rebuild bitmap file

Explanation: Super block container invalid bitmap file inode number.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADCELLSIZE, Problem: Bad container file cell size: <bad value>

Solution: Will be replaced by default size (<good value>)

Explanation: Super block contains invalid value for the Connection file system cell size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADCHKSUM, Problem: Super Block has mismatched cell group checkbytes <CH1><><CH2>

Solution: Will set checkbytes to same value

Explanation: The front checkbyte byte of the super block cell group does not match with its rear checkbyte byte. The super block may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADCGRTYPE, Problem: Wrong super block cell group type: <bad value>

Solution: Will set cell group type to super block type

Explanation: The type field of the super block cell group does not identify itself as super block.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADCFEXTEND, Problem: Bad container file extension size: <bad value>

Solution: Will be replaced by default value (<good value>)

Explanation: Super block contains invalid value for the initial container file extension size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADCGSIZE, Problem: Bad Super Block cell group size: <bad value>

Solution: Will be replaced by correct Super Block size: <good value>

Explanation: The size field of the super block cell group does not correctly describe the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADDCGSIZE, Problem: Bad initial directory cell group size: <bad value>

Solution: Will be replaced by default value (<good value>)

Explanation: Super block contains invalid value for the initial directory cell group allocation size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADECGSIZE, Problem: Bad initial extension cell group size: <bad value>

Solution: Will be replaced by default size (<good value>)

Explanation: Super block contains invalid value for the initial extension cell group allocation size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADFID, Problem: Bad container file FID: <bad FID>

Solution: Will be replaced by correct FID: <good FID>

Explanation: The VMS file identification (FID) of the container file in the super block does not match the real container file FID.

User Action: If the CONFIRM option is set, you will be prompted to confirm the solution before it is implemented.

%UCX$-E-ANA_SUP_BADHIBK, Problem: Bad maximum container file size: <bad value>

Solution: Will be replaced by default value (<good value>)

Explanation: The value in the super block for the maximum size to which the container file can be expanded is invalid.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADIICGSIZE, Problem: Bad initial inode cell group size: <bad value>

Solution: Will be replaced by default size (<good value>)

Explanation: Super block contains invalid value for the initial inode cell group allocation size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADSUPBLK, Problem: Super Block irremediably corrupted. ** End ANALYZE **

Explanation: The super block is totally unrecognizable. ANALYZE /CONTAINER cannot continue the analysis.

User Action: Create a new container file using the UCP command CREATE CONTAINER, delete (or save) the corrupted container file, and run ANALYZE/CONTAINER again. ANALYZE/CONTAINER analyzes the container file and will move the entire directory hierarchy to the lost and found directory. This means the root directory will be under the lost and found directory, but all files and directories under root will be intact.

%UCX$-E-ANA_SUP_BADRTHRESH, Problem: Bad directory relocation threshold: <bad value>

Solution: Will be replaced by default value (<good value>)

Explanation: Super block contains invalid value for the directory's relocation threshold that controls the directory's size.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_SUP_BADVERSION, Problem: Bad container file version: <bad value>. Current version is <good value>

Solution: Will abort ANALYZE/CONTAINER

Explanation: The container file version number in the super block is not a valid version number or a currently supported version number.

User Action: None. ANALYZE/CONTAINER will abort the analysis.

%UCX$-E-ANA_SUP_BADVOL, Problem: Bad volume name: <bad string>

Solution: Will be replaced by: <good string>

Explanation: The VMS volume name of the device on which the container file is located does not match with the real volume name.

User Action: If the CONFIRM option is set, the user will be prompted to confirm the solution before it is implemented.

B.3.1 Inode Validation Messages

The following error messages are related to errors detected in inodes. Generally the problem is corrected automatically (if the REPAIR option is set) without user confirmation or action. Problems can be any of the following:

- File/directory has been copied or restored from backup.
- A non-Connection file system file has been copied into the Connection file system's VMS directory hierarchy, or the file has been renamed to a name that does not conform to the Connection file system convention.
- Data corruptions caused either by a hardware problem or software problem, or the system was shut down abruptly while the Connection file system was active.

If you believe the problem was due to software, please send an SPR.

%UCX-E-ANA_INO_BADCHKSUM, Problem: <file type> inode (<inode number>) has mismatched cell group checkbytes <CHK1> <> <CHK2>

Solution: Will set checkbytes to same value

Explanation: The front checkbytes of the inode cell group do not match with its trailer checkbytes byte. The inode may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADATIME, Problem: <file type> inode (<inode number>) has wrong accessed time: <wrong time>

Solution: Will be replaced by correct time: <correct time>

Explanation: The file accessed time in the inode is less than the actual accessed time.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADBUFSFILENAM, Problem: Bad CFS ULTRIX file system file name: <bad file name>

Explanation: The name of the VMS file in Connection file system does not have the correct Connection file system name format.

User Action: Correct the name of the VMS file.

%UCX-E-ANA_INO_BADCGSIZE, Problem: Bad <file type> inode <inode number> inode cell group size: <bad value>

Solution: Will be replaced by initial inode size (good value)

Explanation: The size field of the inode cell group does not correctly describe the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADCGTYPE, Problem: Bad <file type> inode cell group type: <bad value>

Solution: Will be replaced by correct cell group type: <good value>

Explanation: The type field of the inode cell group does not identify itself as an inode.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADCHKSUMDEL, Problem: <file type> inode (<inode number>) has mismatched checkbytes:<CHK1 <> <CHK2>

Solution: Will be deleted

Explanation: The front checkbytes of the inode cell group do not match with its trailer checkbyte byte. The inode is either a symbolic link or an imported file that may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADCTIME, Problem: <file type> inode has wrong change time: <wrong time>

Solution: Will be replaced by correct time: <correct time>

Explanation: The file's change time attribute in the inode is different from the actual creation time.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADEXTMPTR, Problem: Extension map pointer of <file type> inode (<inode number>) contains the wrong size:

Solution: Will be replaced by correct size: <good value>

Explanation: The inode's extension map pointer contains the wrong cell group size for the extension cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADFID, Problem: <file type> inode contains bad FID: <bad FID>

Solution: Will be replaced by correct FID: <good FID>

Explanation: The VMS file identification (FID) of the file in the file's inode does not match the real FID of the file.

User Action: If the CONFIRM option is set, you will be prompted to confirm the solution before it is implemented.

%UCX-E-ANA_INO_BADFILESIZE, Problem: <file type> inode (<inode number>) contains bad file size: <bad value> bytes

Solution: Will be replaced by the correct size: <good value>

Explanation: The file size field of the inode does not correctly describe the size of the file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADFNAM, Problem: <file type> inode (<inode number>) file name string has invalid characters

Solution: The inode will be deleted

Explanation: The symbolic link or imported file inode contains invalid file name.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADINODE, Problem: Bad <file type> inode <inode number> cannot be recovered

Solution: Will be deleted

Explanation: The inode contains too many errors to be repaired.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADINODEOK, Problem: Bad <file type> inode <inode number>; cannot be recovered

Solution: Will try to rebuild it

Explanation: The inode contains too many errors to be repaired. It (<inode number>) should be deleted and rebuilt.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_INO_BADINUM, Problem: Bad <file type> inode number:/ <bad inode number>

Solution: Will be replaced by the correct inode number: <good inode number>

Explanation: Each inode contains its own inode number. The value in the inode number field of this file is not its inode number.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADMAXMPTR, Problem: Bad <file type inode/extension inode> max. number of mapptrs: <bad value>

Solution: Will be replaced by the correct number: <good value>

Explanation: Each inode and its extensions has a dynamic portion that contains certain number of internal map pointers. The field that describes the limit of map pointers in the dynamic portion is wrong.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADMODE, Problem: <file type> inode contains bad file mode: <bad mode>

Solution: Will be replaced by the correct mode: <good mode>

Explanation: The inode contains wrong file mode (file type and protection bits).

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADMPTR, Problem: Bad map pointer in <file type> inode (cg:<cell number>)

Solution: Will be cleared

Explanation: The inode contains an invalid map pointer to the file data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADMPUSED, Problem: <file type> inode (<inode number>) has bad used map pointers count:

Solution: Will be replaced by the correct count: <good value>

Explanation: Each inode and its extensions has a dynamic portion that contains certain number of internal map pointers. The field that describes the number of map pointers being used is wrong.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADMTIME, Problem: <file type> inode has wrong modification time: <wrong time>

Solution: Will be replaced by correct time: <correct time>

Explanation: The file modified time in the inode is less than the actual modified time.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADNAMLEN, Problem: <file type> inode (<inode number>) has bad filename length: <bad value>

Solution: The inode will be deleted.

Explanation: The symbolic link or imported file inode contains invalid file name length.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADOWNINUM, Problem: <file type> inode (cg:<cell number>) contains bad owning inode number:<bad value>

Solution: Will be replaced by correct owning inode number: <good value>

Explanation: Each inode and its extensions has a dynamic portion that contains the inode number of the inode. The inode number in the dynamic portion does not match the inode's real inode number.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADPTRSUP, Problem: Bad Super Block inode map pointer: <bad map pointer>

Solution: Will be replaced by map pointer to super block: <good map pointer>

Explanation: The super block inode does contains bad map pointer to the super block.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADSEGNO, Problem: Bad <file type> inode (<inode number>) segment number: <bad seg value>

Solution: Will be replaced by correct segment number: <good value>

Explanation: Each inode and its extensions has a dynamic portion that is enumerated sequentially by its segment number. The segment number is out of sequence.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_BADTYPE, Problem: Bad <file type> inode (<inode number>) type: <bad type>

Solution: Will be replaced by correct inode type: <good type>

Explanation: The type field in the inode does not correctly describe the type of the file that owns the inode.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BADVMSDIR, Problem: Cannot read VMS directory <directory name>, probably corrupted

Solution: Directory and its subdirectories will be removed from container file

Explanation: The VMS directory in the CFS ULTRIX file system is not readable. It is probably corrupted. ANALYZE/DISK is recommended.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_INO_NOVMSDIR, Problem: VMS directory inconsistency: directory <directory name> not found in container directory

Solution: Will remove the directory

Explanation: The VMS container directory contains an entry for the directory, which does not exist on disk. ANALYZE/DISK is recommended.

User Action: If the CONFIRM option is set, the user will be prompted to confirm the solution before it is implemented.

%UCX-E-ANA_INO_NOVMSFILE, Problem: VMS directory inconsistency: file <file name> not found in directory <directory name>

Solution: Will remove the file from VMS directory.

Explanation: The VMS container directory contains no entry for the file, which does not exist on disk. ANALYZE/DISK is recommended.

User Action: If the CONFIRM option is set, the user will be prompted to confirm the solution before it is implemented.

%UCX-E-ANA_INO_NULVMSOWNER, Problem: File/directory has invalid owner UIC: [XX,XX]

Solution: Will replaced by the parent directory owner UIC.

Explanation: A VMS directory or file in the container file system has an invalid UIC.

User Action: If the CONFIRM option is set, the user will be prompted to confirm the solution before it is implemented.

B.4 Directory and Raw Data Validation Messages

The following error messages are related to errors detected in directories and raw data cell groups (internal use only). All these problems are corrected automatically (if the REPAIR option is set) without user confirmation or action.

The errors indicate data corruptions caused either by a hardware problem or software problem, or the system was shut down abruptly while the Connection file system was active. If you believe the problem was due to a software problem, please send an SPR. If a directory is corrupted and cannot be repaired, all the files and directories under it will be moved to the lost and found directory.

%UCX-E-ANA_DIR_BADALLOCERR, Problem: Directory cel group <cell pointer> allocation error: duplication of out of container file size

Solution: Will be cleared

Explanation: The directory claims ownership of space already allocated to other files, or the space is out of the file system limits.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADCGSIZE, Problem: Bad directory <cell pointer> cell group size: <bad value>

Solution: Replace by initial directory size: <good value>

Explanation: The size field of the directory cell group does not describe correctly the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADCGTYPE, Problem: Bad directory cell group <cell pointer> type: <bad type>

Solution: Will be replaced by the correct type: <good type>

Explanation: The type field of the directory cell group does not identify itself as a directory.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADCHKSUM, Problem: Directory <cell pointer> has mismatched cell group checkbyte: <CHK1> <> <CHK2>

Solution: Will set checkbytes to same value

Explanation: The front checkbyte of the directory cell group does not match with its rear checkbytes byte. The directory may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADDIRSIZE, Problem: Directory inode <inode number>contains bad file size: <bad value> bytes

Solution: Replace by the correct size: <good value>

Explanation: The directory's inode contains wrong size of the directory in the container file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADDOTLEN, Problem: Bad <. or ..> entry length: <bad value> (cg: <cell number>)

Solution: Replace by correct length: <good value>

Explanation: The '.' or '..' entry contains wrong total entry length.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADDOTNAM, Problem: Directory <cell pointer> has bad <. or ..> entry name

Solution: Replace by <. or ..>

Explanation: The directory entry should have '.' or '..' as the file name.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADDOTNAMLEN, Problem: Bad <. or ..> entry name length: <bad value> (cg: <cell number>)

Solution: Replace by correct length: <good values>

Explanation: The directory's '.' or '..' entry has wrong file name length (different to 1 or 2).

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADDOTNUM, Problem: Bad <. or ..> entry inode number: <bad inum> (cg: <cell number>)

Solution: Replace by correct inode number: <good inum>

Explanation: The '.' or '..' entry contains inode number that does not correspond to the inode number of the directories '.' or '..'.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADENTNAME, Problem: Bad directory <cell pointer> entry name: <up to 10 chars of name>

Solution: Will be cleared

Explanation: The directory entry contains invalid file name string (up to 10 characters of the file name are printed).

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADENTNAMLEN, Problem: Bad directory <cell pointer> entry name length: <bad value>

Solution: Will be replaced by the correct length: <good value>

Explanation: The directory entry contains invalid file name length.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX$-E-ANA_DIR_BADENTRYLEN, Problem: Bad directory (<cell group number>) entry length: <bad value>

Solution: Will be replaced by the correct length: <good value>

Explanation: The directory entry container invalid total entry length.

User Action: None. The error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADHIERARCHY, Problem: Directory hierarchy corrupted: multiple directory tree found!

Solution: Will place files not connected to root in lost and found directory

Explanation: The directory hierarchy does not form a unique tree structure. The hierarchy is broken into disconnected trees. Some files or directories lost the link with their parent directories.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADMAXERROS, Problem: Bad directory (cg: <cell group>); cannot be recovered

Solution: Will be deleted. All its files will be moved to lost and found

Explanation: The directory is completely corrupted and cannot be repaired. All the file and directories under this directory will be moved to the lost and found directory.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADPNORAWSIZE, Problem: Raw data inode has bad file size: <bad value> bytes

Solution: Replace by correct size: <good value>

Explanation: The raw data's inode contains wrong size of the raw data file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADRAWCG, Problem: Bad raw data file <cg pointer> (cannot be corrected)

Solution: Will be deleted

Explanation: The raw data file is completely corrupted and cannot be repaired.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADRAWCHKSUM, Problem: Raw data <cg pointer> has mismatched cell group checkbyte: <CHK1> <> <CHK2>

Solution: Will set checkbytes to same value

Explanation: The front checkbyte of the raw data cell group does not match with its rear checkbytes byte. The raw data file may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADRAWSIZE, Problem: Raw data cell group <cg pointer> has bad size: <bad value>

Solution: Replace by correct size: <good value>

Explanation: The size field of the raw data cell group does not correctly (<inode number>) describe the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADRAWTYPE, Problem: Raw data cell group <cell pointer> has bad type: <bad type>

Solution: Will be replaced by correct type: <good type>

Explanation: The type field of the raw data cell group does not identify itself as a raw data file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_DIR_BADROOT, Problem: Root directory corrupted

Solution: Will be re-created. Its files will be placed in lost and found

Explanation: The root directory is completely corrupted. A new root directory will be created, and all the files and directories under the old root directory will be moved to the lost and found directory.

User Action: None. Error will be repaired automatically if REPAIR option is set.

B.5 Bitmap and Deleted Cells Validation Messages

The following error messages are related to errors detected in directories and raw data cell groups (internal use only). All problems are corrected automatically (if the REPAIR option is set) without user confirmation or action. The errors indicate data corruptions caused either by a hardware problem or software problem, or the system was shut down abruptly while the Connection file system was active. If you believe the problem was due to software problems, send an SPR.

%UCX-E-ANA_BMP_BADBITMAP, Problem: Bitmap allocation from cell <cell number> to cell (cell number) <bad value>

Solution: Will be reallocated

Explanation: The bitmap file has wrong allocation information for the cell (bitmap file indicates the cell as allocated but it is actually free).

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADBCGSIZE, Problem: Bad bitmap cell group allocation: <inode number>

Solution: Will be rebuilt

Explanation: The bitmap cell group claims space already allocated to other files.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADBMPCGRPSIZE, Problem: Bad bitmap cell group (cg: <cell number>) cell group size: <bad value>

Solution: Replace by correct value: <good value>

Explanation: The size field of the bitmap cell group does not correctly describe the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADBMPCGRPCHKSUM, Problem: Bitmap cell group (cg: <cell number>) has mismatched cell group checkbyte: <CHK1> <> <CHK2>

Solution: Will set checkbytes to same value

Explanation: The front checkbyte of the bitmap cell group does not match with its rear checkbytes byte. The bitmap file may contain corrupted data.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADBMPCGRPTYPE, Problem: Bad bitmap cell group (cg: <cell number>) cell group type: <bad type>

Solution: Replace by correct type: <good type>

Explanation: The type field of the bitmap cell group does not identify itself as a bitmap.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADDCGRPCHKSUM, Problem: Deleted cell (cg: <cell number>) has mismatched cell group checkbyte

Solution: Will set checkbytes to same value

Explanation: The front checkbyte of the deleted cell group does not match with its rear checkbytes byte.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADDCGRPS, Problem: Cells <cell number> to <cell number> should be deleted cells but have wrong attributes

Solution: Will be corrected to be deleted cells

Explanation: The group of cells are allocated as deleted cells but do not have the correct attributes.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADDCGRPSIZE, Problem: Deleted cell (cg: <cell number>) has bad cell group size: <bad value>

Solution: Replace by correct value: <good value>

Explanation: The size field of the deleted cell group does not correctly describe the size of the cell group.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_BADDCGRPTYPE, Problem: Deleted cell (cg: <cell number>) has bad cell group type: <bad type>

Solution: Replace by correct type: <good type>

Explanation: The type field of the deleted cell group does not identify itself as a deleted cell.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-E-ANA_BMP_NEWBMP, New bitmap file created

Explanation: Due to problems detected in the bitmap file, a new bitmap file has been created.

User Action: None.

%UCX-E-ANA_BMP_BADNULLMPTR, Problem: Unused bitmap file map pointer has bad value: <bad map pointer>

Solution: Will be replaced by correct value: 00000000

Explanation: Unused map pointers in the bitmap inode contains non-null value.

User Action: None. Error will be repaired automatically if REPAIR option is set.

B.6 Directory Hierarchy Reconstruction Messages

The following error messages are produced during the reconstruction of the directory hierarchy. All problems are corrected automatically (if the REPAIR option is set) without user confirmation or action. The problem indicates data corruptions caused either by a hardware or software problem, or the system shutting down abruptly while the Connection file system was active. If you believe the problem was due to software problems please send an SPR.

%UCX-I-ANA_FIN_BADNLINK, Problem: Bad inode <inode number> link count: <wrong value>

Solution: Will be replaced by correct number: <correct value>

Explanation: Inode contains wrong number of links to this file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-I-ANA_FIN_FORIEGN, Problem: Cannot delete bad directory: contains non-CFS ULTRIX file system files

Solution: Directory will be removed from ULTRIX file system but not deleted

Explanation: Non-Connection file system files have been placed in Connection file system directory. As a result, the directory cannot be deleted.

User Action: Remove the non-Connection file system files from the Connection file system directories and run ANALYZE/CONTAINER again.

%UCX-I-ANA_FIN_NOBMP, Problem: Bitmap file entry is not found in the root directory

Solution: Will add the bitmap file entry to the root directory

Explanation: The root directory does not have an entry for the bitmap file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

%UCX-I-ANA_FIN_NOSUPBLK, Problem: Super Block entry is not found in the root directory

Solution: Will add the Super Block entry to the root directory

Explanation: The root directory does not have an entry for the Super Block file.

User Action: None. Error will be repaired automatically if REPAIR option is set.

Glossary

ACK

A control bit (acknowledgment flag) in the TCP header that indicates that the acknowledgment number field is significant for this segment.

Absolute pathname

A pathname that starts with a slash (/) and specifies a file that can be found by starting at the root of the file system and traversing the file tree.

Active port

A port that is bound to a process.

Address Resolution Protocol (ARP)

A protocol that dynamically maps between Internet addresses and Ethernet addresses.

Alternate address notation

An Internet address notation that conveys the same information as the common notation, but consists of two parts: network and host.

Application layer

The highest layer in the Internet architecture model that provides the application services. Examples of application services would be network file service (NFS), file transfer protocol (FTP), and MAIL.

Berkeley Internet Name Domain (BIND)

A host name and address lookup service for the Internet network. The BIND service is implemented in a client-server model. The client software (implemented by the Connection), is referred to as the resolver. The resolver allows client systems to obtain host names and addresses from servers rather than from locally hosted databases. As a result, you can use the BIND service to replace or supplement the host address mapping provided by the local UCX$HOST file.

Binding

Defining an ULTRIX or VMS file system to be a part of the Connection file system.

Bound port

A port is bound to a process by an I/O function that specifies a port number and Internet address for the device-socket.

CFSRTL

A VMS run-time library (RTL) that is used by the NFS server process to process files in the Connection file systems.

Checksum

A parameter in the header whose value can be used to determine whether or not the data was corrupted over the network.

Client

A process that sends a request and waits for the results from another process that offers a service over the network.

Common address notation

The common way of notating an Internet address. The 32-bit address uses four fields that are separated by periods and each field ranges from 0 to 255.

Connection

A logical communication path between two different processes.

Connection file system (CFS)

A collection of ULTRIX and VMS file systems organized as a single-level hierarchy.

Connection VMS server

A computer system that offers services to workstation clients within an Internet network environment. The computer system can be a single host, a whole VAXcluster system, or members of a VAXcluster system.

Container file

An RMS data file that contains an ULTRIX directory structure and ULTRIX file attributes for an ULTRIX file system. Each ULTRIX regular file is stored as a separate RMS data file using a system-assigned valid file name. The directory data files in the container file contain the ULTRIX file names and a pointer to the corresponding Files–11 data file.

Datagram

A self-contained package of data carrying enough information to be routed from source to destination without reliance on earlier exchanges between source or destination and the transporting network.

Datagram fragment

The result of fragmenting a datagram. The fragment carries a portion of data from the larger original and a copy of the original datagram header. The header fragmentation fields are adjusted to indicate the fragment's relative position within the original datagram. (See also, datagram and fragmentation.)

Datagram service

A datagram that is delivered in such a way that the receiver can determine the boundaries of the datagram as it was entered by the source.

Device-socket

An extension of the pseudodevice used for communications. The device-socket consists of the Internet pseudodevice and socket.

Destination address

An Internet address that specifies where a datagram has to be sent. It contains the network, host identifiers, and eventually the subnet identifier.

Destination port

A 2-octet value in the TCP and UDP header field that identifies the destination upper level protocol for a segment's data.

Exported file

A file with an ULTRIX file system that has been copied or linked into a VMS file system.

Exporting a file system

Identifying a Connection file system or directory that can be remotely mounted by NFS clients.

File system

A method for recording, cataloging, and accessing files on a volume.

Files–11 ODS level 2 structure

The set of rules that govern the organization of the disk external to files.

File Transfer Protocol (FTP)

A protocol that allows users to log in to a remote host, identify themselves, list remote directories, copy files to or from the remote host, and execute a few simple commands remotely.

Fragmentation

The breaking up of Internet datagrams into smaller datagrams. This allows a datagram that originates in a local network that allows a large packet size to tranverse to a local network that limits packets to a smaller size. The Internet fragments are reassembled at the destination host. (See also datagram fragment.)

Gateway

A host computer that interconnects two networks and transfers packets from one network to another.

Hard link

A link to a file that is indistinguishable from the original directory entry. (See link.)

Header

A collection of control information transmitted with data between peer entities.

Host

A computer that is a source or destination of messages of the communication subnetwork (referred to as a node in VMS terminology).

Host database

An Internet database that allows users to use host names. The database contains host names, Internet addresses of the hosts, and any alias names for the host.

Host-to-host layer

The second-highest level in the Internet architecture model. This layer provides end-to-end communication services, including mechanisms such as end-to-end reliability and network control. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) reside in this layer.

IHL (Internet header length)

An IP header field that indicates the number of 32-bit words making up the Internet header.

Imported file

A file within a VMS file system that has been copied or linked into an ULTRIX file system.

INS (Initial sequence number)

The first sequence number used for sending or receiving on a connection.

Internet address

A 32-bit address that is composed of two parts: network number and host number.

Internet architecture

A four-layered communications model that consists of the following layers: application layer, host-to-host layer, Internet protocol layer, and network protocol layer.

Internet Control Message Protocol (ICMP)

A special-purpose protocol that gateways use to communicate with the network software in hosts.

Internet datagram

The package exchanged between a pair of IP modules. The datagram consists of an IP header and data portion.

Internet network

A collective network of cooperative, interconnected networks. The hosts on the same network are physically interconnected and the networks are physically interconnected by a host known as a gateway.

Internet Protocol (IP)

A protocol that resides in the Internet layer and performs two major functions: Internetwork addressing and fragmentation of messages.

Internet pseudodevice

A VMS pseudodevice that provides the mechanism for the VMS user to interface with the Internet protocols.

Link

A directory entry referring to a file. A file can have several links to it. A file cannot be deleted (removed) until the link count is 0. (See also hard link.)

Local address

The address of a host within a subnetwork.

Local area network

Consists of two or more computer systems connected by an Ethernet communication medium. Each host computer connects to the transmission medium by a hardware interface connected to only one local area network.

Local network

The network directly attached to a host or gateway.

Local subnetwork

The subnetwork directly attached to a host or gateway.

Loopback

A test of connectivity to a specific host in the network.

MBUF

A mnemonic for the term memory buffers.

Network class

Defines the type of network addressing scheme being used. The high-order bits in the network number designate the network class of the Internet address.

Network class

Defines the type of network addressing scheme being used. The high-order bits in the network number designate the network class of the Internet address.

Network database

A database that allows users to refer to networks by name rather than network number. The database contains network names, Internet addresses for the networks, and any alias names for the networks.

Network File System (NFS)

A server that provides transparent remote access to files on a VMS server for disk-based UNIX clients.

Network layer

The lowest layer in the Internet architecture model. This layer provides the mechanism for connecting the hosts to the networks.

NFS server

A VMS image that simultaneously processes multiple service requests from many NFS clients. The services provided are MOUNT, NFS, and PORTMAPPER.

Nobody

A UNIX convention, used when file ownership is not known, that maps to an account with a UID and GID of –2.

Pathname

A UNIX pathname is composed of a series of fields separated by slashes (/). Each field designates a file name that is uniquely contained in the previous field (directory).

Peer socket

The socket on the remote host.

Port

The endpoint of a communication link between two processes.

Privileged port

A port in which the remote host has done some level of checking against the application in using the port. Privileged port numbers range from 1 to 1023.

Reassembly

The process of piecing together datagram fragments to reproduce the original large datagram. Reassembly is based upon the fragmentation data in the IP header of the datagram.

Reliability

The ability of a protocol to recover data that is damaged, lost, duplicated, or delivered out of order.

Relative pathname

A pathname that does not start at the root. The default directory is merged with the relative pathname to form the absolute pathname.

Reserved port

An assigned port that provides services to unknown callers by providing a service contact point. Reserved port numbers range from 1 to 255.

RMS

The data management subsystem on VMS that defines the rules that govern the internal organization and the methods of accessing file data. VAX RMS together with ODS-2 define a set of rules that govern files in a VMS File System. These rules include how files are named and how files are cataloged in directories.

Root

The element of a pathname that identifies the target file system.

Route database

A database that contains routing information. The database contains destination host names, Internet addresses for the hosts, gateway host names, and Internet addresses for the gateways. There are two logical route databases: the static route database that is maintained on-disk and the volatile database in memory.

Routing Information Protocol (RIP)

A protocol that enables gateways to broadcast their current routing database to host and networks that are connected directly to them. The Connection implements the RIP protocol through its dynamic routing server.

Segment

The unit of data exchanged by the TCP modules.

Segment length

The amount of sequence number space occupied by a segment, including any controls that occupy sequence space.

Sequence number

A 32-bit field in the TCP header that contains the sequence number of a sequenced control flag, the first byte of data, or empty segments (the sequence number of the next data octet to be sent).

Server

A process that offers a service over the network to another process. A server accepts requests from other processes known as clients.

Socket

The endpoint of a communication to which an Internet address and port may be bound.

Source

An IP header field that contains the Internet of the datagram's point of origin.

Source port

A 2-octet value in the TCP or UDP header field that identifies the source's upper-level protocol of a segment's data.

Subnet mask

A mask used to determine the subnetwork in the Internet address. Each bit that is turned on (binary 1) in the mask is interpreted as part of the network and subnetwork address.

Subnetwork

A group of hosts within a network into logical groups. A network can be made up of several subnetworks.

Superuser

A user who has been granted special privileges. A superuser has an effective UID of 0.

Symbolic link

A special type of file that contains the name of the file to which it is linked. The referenced file is used whenever the symbolic link file is opened.

Telnet protocol

A protocol enables users to access any system on a network running the Telnet server software and establishes a virtual terminal connection between their terminals and the specified hosts.

Thread

A request from an NFS client to the NFS server.

Transmission Control Protocol (TCP)

A reliably delivered full-duplex stream-oriented protocol that supports demultiplexing based on a port number. TCP allows individual processes to establish stream connections without interfering with each other.

ULTRIX file system

A collection of files organized as a tree, with a single root node called "root" which is written as a slash (/). Non-leaf nodes being directory files and leaf nodes are either directory or regular data files. On a Files–11 ODS-2 formatted disk, an ULTRIX file system is represented as a set of Files–11 files.

User Datagram Protocol (UDP)

An unreliable delivered protocol that depends on the underlying Internet Protocol to transport UDP messages from one host to another. Each UDP message contains the data sent by a user process, a destination port number, and a source port number.

VAXcluster alias

An alias that allows remote hosts to address the cluster of hosts as a single host, as well as any cluster member individually.

VMS file system

The VMS files and directories on a mounted VMS volume. These files and directories reside on a Files–11 On-Disk Structured (ODS-2) disk.

Well-known port

An assigned port that provides services to unknown callers by providing a service contact point. Reserved port numbers range from 1 to 255.

Window

A 2-octet field in the TCP header that indicates the number of data octets (relative to the acknowledgment number in the header) that the sender is currently willing to accept.

Index

A

Addressing
  See Internet address
Address notation
  See Internet address
Address resolution protocol
  description of, 2–6
  maps Ethernet/Internet addresses, 3–26
  when to use, 3–26

B

Berkeley Internet Name Domain resolver, 1–3, 2–8 to 2–9, 3–13
  configuring, 3–7 to 3–8
  enabling, 3–7 to 3–8
  managing the, 3–36
  troubleshooting, 3–42 to 3–43
Bind
  VMS files, 4–11
BIND
  See Berkeley Internet Name Domain resolver
Binding
  a container file, 4–11
  a file system, 4–10
  requirements for, 4–11
Binding ports
Binding ports (Cont.)
  privileges required for, 2–4
Broadcast mask
  default value of, 2–15
  entering information for, 3–6
  purpose of, 2–14
Broadcast messages, 3–20
Bytes
  displaying number of transmitted and received, 4–33

C

Cache size of host
  increasing, 3–2
CFS ULTRIX file system, 4–5
  components of, 4–18
  creating an, 4–10
  integrity of, 4–17
CFS VMS file system, 4–4
Channels
  See NFS server
  quota value for, 4–33
Checksum
  See Internet performance
  for TCP and UDP, 3–20
client
  See Process
Client access
  See proxy database

Client-server model, 2–3
Connection databases, 5–3
Connection file system, 4–6
Connection VMS server
  vaxcluster used as, 2–2
Container file, 4–5
  creating an, 4–11
  number of, 4–11
  removing, 4–16

D

Databases
  See also Internet database
  EXPORT, 3–4, 5–3
  HOST, 3–3, 3–5, 3–13, 3–13 to 3–14, 5–3
  NETWORK, 3–3, 3–5, 3–13, 3–14 to 3–15, 5–3
  PROXY, 3–4, 5–3
  ROUTE, 2–6, 3–3, 3–5, 3–13, 3–15, 5–3
Datagram fragment
  See Fragmentation
  reassembling a, 2–19
DCL tables
  becoming corrupted, 3–3
Device-sockets
  consists of, 3–19
Directory
  removing ULTRIX, 4–16
Displaying
  See also NFS server
  Internet control/statistical information, 3–26
  specific host or client information, 4–31
Dynamic routing, 2–6
  See also Routing Information Protocol
  configuring, 3–8 to 3–9
  enabling, 3–8 to 3–9
  managing, 3–35 to 3–36

E

Error messages
  displaying number of, 4–33
  mapping VMS error messages into NFS, 4–40
Errors
  displaying number of, 4–32
Ethernet controller
  entering information for, 3–6
  found by UCX$CONFIG.COM, 3–5
  name for, 3–5
Export database
  creating, 4–7
Exporting
  CFS VMS file system, 4–10
  file systems and directories, 4–1
  the mount point, 4–10
  UNIX file system, 4–13

F

Files
  changing permission for ULTRIX, 4–12
  classes of users, 4–12
  closing, 4–26, 4–30
  copied between file systems, 4–16
  displaying number of opened, 4–33
  limit, 4–29
  maximum number of accesses to, 4–30
  ownership of, 4–26
  permissions for, 4–12
  timestamp for opened, 4–27
  translating owner's identity of, 4–8
File transfer protocol
  description of, 2–8
File Transfer Protocol
  logical names, 3–37
  managing, 3–36 to 3–38
  shutting down the server, 3–37
  starting the server, 3–36
Fragmentation
  by gateway, 2–19

Fragmentation (Cont.)
of datagram, 2–19

G

Gateway, 3–16 to 3–17
See Fragmentation
definition of, 2–1
Internet address for, 2–16
Global pages
See Internet parameter
increase, 3–3
required number of, 3–3
Global sections
See Internet parameter
increasing, 3–3
required number of, 3–3

H

Host command procedure
See Internet logical name, 3–13
Host database
contents of, 3–13
making entries in, 3–14
Host name
entering information for, 3–6
Host names
UNIX, 3–13
Hosts in VAXclusters, 3–11 to 3–12

I

Interface type
supported, 3–23
Internet address
address notation, 2–10
alternate notation for, 2–10
common notation for, 2–10
definition of, 2–2, 2–3
entering information for, 3–6
for routing, 2–16
obtaining, 2–12, 3–5
parts of, 2–9
Internet command procedure
Internet command procedure (Cont.)
UCX$CONFIG.COM, 3–3, 3–13
UCX$HOST.DAT, 3–13
UCX$INET_SET_INTERFACES.COM
, 3–4, 3–18, 3–23, 3–26
UCX$INET_SHUTDOWN.COM,
3–12
UCX$INET_START.COM, 3–12
Internet communications
shutting down, 3–12
starting, 3–23, 3–26
Internet concepts, 2–1 to 2–4
client-server model, 2–3
overview, 2–1 to 2–3
ports, 2–3
Internet control buffers
See Internet mbufs
Internet control message protocol
description of, 2–5
Internet database, 3–13 to 3–15
See host database
See network database
See proxy database
See route database
Internet data buffers
See Internet mbufs
Internet file
SYS$SYSTEM:MODPARAMS.DAT,
3–2, 3–3
UCX$HOST.DAT, 3–13
UCX$NETWORK.DAT, 3–14
UCX$ROUTE.DAT, 3–15
Internet interface
creating, 3–23, 3–26
entering information for, 3–6
name of, 3–23
pseudo-interface, 3–23 to 3–24
Internet logical name
UCX$HOST.DAT, 3–13
UCX$NETWORK.DAT, 3–15
Internet mbufs
calculating number of dynamic, 3–19
dynamic allocation of, 3–19
space requirements for, 3–18

Internet mbufs (Cont.)
  static allocation of, 3–18
Internet message
  for configuration process completion, 3–9
  for finding Ethernet controller, 3–5
  for host name of found Ethernet controller, 3–6
Internet network
  definition of, 2–1
  setting-up, 3–1, 3–3
Internet parameter, 3–2 to 3–4
  GBLPAGES, 3–3
  GBLSECTIONS, 3–3
  NPAGEDYN, 3–2
  NPAGEVIR, 3–2
Internet performance
  affects of IP reassembly time on, 3–21
  disabling checksum for improved, 3–20
Internet privileges
  for UCX$CONFIG.COM, 3–4
Internet Protocol
  routing messages by, 2–16
Internet protocols
  See also Address resolution protocol
  See also File transfer protocol
  See also Internet control message protocol
  See also Internet protocol
  See also Transmission control protocol
  See also User datagram protocol
  description of, 2–5
Internet pseudodevice
  assigning a channel to, 3–19
  creating, 3–19
Internet software
  discards network messages, 3–21
  requires UCX$ROUTE.DAT, 3–15
  starting, 3–3, 3–17
IP reassembly time
  definition of, 3–21

L

Links, 4–16
  removing, 4–16
Local area network
  definition of, 2–1

M

Memory
  controlling size of virtual, 4–30
  displaying allocation of, 4–32
  wasting virtual, 4–26
Messages
  discarding, 3–21
  displaying duplicate/dropped, 4–32
Mount
  a remote directory, 4–1
  done by superuser, 4–1
  requirements to, 4–7
  service, 4–7
Multi-homed hosts, 3–9 to 3–11

N

Network
  configuring, 3–5 to 3–12
    hosts in VAXclusters, 3–11 to 3–12
    multi-homed hosts, 3–9 to 3–11
Network database
  contents of, 3–14
Network File System
  backing up files, 4–18
  restoring files, 4–19
    ULTRIX file systems, 4–19
Network mask, 2–12
  entering information for, 3–6
Network routing, 2–16
Networks
  class a, 2–11
  class b, 2–11
  class c, 2–11, 2–12
  description for classes of, 2–10
  when to use a class a, 2–12

Networks (Cont.)
when to use class b, 2–12
NFS client
accessing a NFS server, 4–5
authenticating, 4–13
creating top-level directory for, 4–12
file quota of, 4–31
giving VMS identity to, 4–8
removing access privileges of, 4–8
setting-up file system for, 4–10
timeout, 4–28
VMS operating system uses identity of, 4–8
NFS logicals
UCX$NFS00000000_ERROR, 4–25
UCX$NFS00000000_GID, 4–26
UCX$NFS00000000_HOSTS, 4–26
UCX$NFS00000000_INACTIVITY, 4–27
UCX$NFS00000000_OPCOM, 4–25
UCX$NFS00000000_OUTPUT, 4–25
UCX$NFS00000000_THREADS, 4–27
UCX$NFS00000000_UID, 4–26
UCX$NFS00000000_XID, 4–28
UCX$NFS_TIME_DIFFERENTIAL, 4–25
NFS operations
displaying number of, 4–33
NFS parameters
CHANNELCNT, 4–30
/EXTENT, 4–30
FILE_LIMIT, 4–29
FILLM, 4–30, 4–31
/PAGE_FILE, 4–30
/WORKING_SET, 4–30
NFS server
authorizing client requests to, 4–8
channels for, 4–29
closing files on, 4–26
defining maximum number of clients for, 4–26
displaying tuning information for, 4–31
NFS server (Cont.)
logging error messages for, 4–25
number of threads for, 4–27
opened files affect performance of, 4–33
performance indicator for, 4–33
performance of, 4–27, 4–30
permission checking of, 4–8
requests to, 4–28
services of, 4–3
starting, 4–20
stopping, 4–37
timer interval for opened files on, 4–26
uses GIDs and UIDs, 4–8
with no proxy account, 4–8
NFS software
starting, 3–4
NFS tuning parameters, 4–21 to 4–36
See also NFS server
Nonpaged dynamic pool
See device-sockets
See Internet mbufs
buffers of, 3–18
controlling, 3–21
device-sockets space requirements for, 3–19
increasing, 3–2

P

Ports
being endpoints of communication, 2–3
communicating between, 2–3
privileged, 2–3
reserved, 2–4
Privileges
See Internet privileges
Process
client, 2–3
definition of, 2–3
server, 2–3

Proxy account
contents of, 4–8
Proxy database
controlling client access with, 4–8
host table build from, 4–26
residing on a VAXcluster system, 4–8
too many client host records for, 4–26
translating from UNIX uid to VMS user name, 4–31

Q

Quota
for device-sockets, 3–21
for send/receive buffers, 3–21
queue size for, 3–21

R

Route, 2–16
Route database
contents of, 3–15
making entries in, 3–15
Routing, 2–16
description of, 2–16
mapping Internet addresses to local addresses for, 2–16
mapping local network addresses for, 2–16
mapping names to addresses for, 2–16
supported, 2–2
Routing Information Protocol, 2–6
See also Dynamic routing

S

Security
granting privileges for, 4–13
restricting network access, 4–13
using mount points for, 4–13
using uid and gid for, 4–13
server
See Process
SET COMMUNICATION /[NO]BROADCAST qualifier, 3–20
SHOW ARP command, 3–34
SHOW COMMUNICATION command, 3–27
SHOW DEVICE_SOCKET command, 3–32
SHOW HOST command, 3–33
SHOW INTERFACE command, 3–31
SHOW PROTOCOL command, 3–29
Starting
See NFS server
Stopping
See NFS server
Subnet mask
See Subnet routing
Subnet routing
class c is not used with, 2–11
description of, 2–18
extending, 3–24 to 3–25
fields of Internet address for, 2–12
using a subnet mask for, 2–12
Subnetwork
definition of, 2–2

T

Telnet, 2–8
system management, 3–38 to 3–42
Internet parameters, 3–40 to 3–41
monitoring, 3–41
server parameters, 3–38 to 3–40
Threads
affect NFS server performance, 4–32
displaying number of, 4–32
Time
Greenwich mean, 4–25
Time interval
for VAXcluster system, 3–25
Transmission control protocol
description of, 2–5
Tuning Internet software, 3–17 to 3–22

U

UCX$CONFIG.COM
  executing, 3–3
  finds Ethernet controller, 3–5
UCX$EXPORT database, 5–3
UCX$HOST.DAT
  See Internet file
UCX$HOST database, 5–3
UCX$NETWORK database, 5–3
UCX$PROXY database, 5–3
UCX$ROUTE database, 5–3
ULTRIX file system
  See CFS ULTRIX file system
User Datagram protocol
  description of, 2–5

V

VAXcluster alias
  ARP used with, 3–25
  for VMS server, 3–25
VMS file system
  See CFS VMS file system
VMS server
  specifying a VAXcluster for, 3–25
  specifying a VAXcluster system for, 3–25
  VAXcluster system used as, 3–25

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

VMS/ULTRIX Connection
System Manager's Guide
AA-LU50C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

What do you like best about this manual? _____

_____

What do you like least about this manual? _____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

---- Do Not Tear – Fold Here and Tape ----

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Publications Manager
Open Software Publications Group
ZK03–2/Z04
110 SPIT BROOK ROAD
NASHUA, NH 03062–9987

--- Do Not Tear – Fold Here ----

Cut
Along
Dotted
Line

Reader's Comments

VMS/ULTRIX Connection
System Manager's Guide
AA-LU50C-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

What do you like best about this manual? _____

_____

What do you like least about this manual? _____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

---- Do Not Tear – Fold Here and Tape ----

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Publications Manager
Open Software Publications Group
ZK03–2/Z04
110 SPIT BROOK ROAD
NASHUA, NH 03062–9987

--- Do Not Tear – Fold Here ----

Cut
Along
Dotted
Line

digital